UNIVERSIDADE DA BEIRA INTERIOR
Engenharia

Wireless Sensor Networking Applied to Swarms of Aquatic Drones

Aleksandra Nadziejko

Dissertação para obtenção do Grau de Mestre em

Engenharia Electrotécnica e de Computadores

(2º ciclo de estudos)

Orientador: Prof. Doutor Fernando José da Silva Velez

Covilhã, Junho de 2016

This dissertation is dedicated to my beloved Family.

Acknowledgements

Dozens of people have contributed to the development process of this dissertation. I own my gratitude to everybody, who believed in me.

I would like to express my gratitude to my supervisor Professor Fernando José da Silva Velez, whose support, expertise, ideas and constructive discussions were the key to achieve the goals of this research.

I wish to thank to Dr. Fardin Derogarian for taking his time to guide me in tangled field of electronics and communications. I own gratitude to Mr. Rui Barata, who provided me technical support and became a mentor in printing circuits.

A very important thanks goes out to Agnieszka Ziebura and Sofia Sousa, without whose motivation and encouragement, I would not achieve what I have achieved today.

A special group of people deserve own part: Prof. Anders Christensen, Prof. Sancho Oliveira, Miguel Duarte, Vasco Costa, Tiago Rodrigues, from BioMachines Lab, ISCTE Lisbon, Prof. Susana Sargento from University of Aveiro, and other team-members from HANCAD project, for welcoming me in the project and for given opportunity to gain engineering experience in the field of communications and robotics.

I am grateful to Prof. Carlos Salema for giving me an opportunity to do research in Instituto de Telecomunicações and his invaluable support in difficult moments. I need to thank to both Instituto de Telecomunicações and University of Beira Interior, which provided me necessary resources facilitating development of this dissertation. In particular this work was partially supported by UID/50008/2013 grant in HANCAD project, ECOOP, and CREaTION, all from Instituto de Telecomunicações.

I wish to thank to my friends at IT-Covilhã, Emanuel Teixeira and Rui Paulo, for their optimism, friendship and help in a variety of matters.

I would like to express my gratitude to my beloved Diogo, who helped me during the experiments and supported me all the time.

At least, but most of all, I wish to thank to my amazing family, polish and portuguese one. You are always there for me. You made it possible to make my dreams come true.

Abstract

Aquatic Unmanned Surface Vehicles (USV) have potential in a variety of maritime activities such as environmental monitoring or sea-life tracking. They can be applied to military missions supporting army in potentially dangerous situations such as reconnaissance or surveillance. USV are capable of many tasks due to technological progress and minimization of equipment in recent years. Size and price drop while reliability improved enables development of large scale multi-agent systems consisting of autonomous USVs.

Multi-agent system of aquatic autonomous USVs may act like a distributed sensing system improving the overall performance when compared to the performance of one unit of USV: more units in the system, larger monitored area. A promising approach inspired from nature is swarm intelligence, which can be found for instance in population of insects such as ants. Swarm behaviour is a motion of large number of units, where each one is autonomous but only as a group they are able to solve the problem. The exchange of information between units is essential for the success of the group. Wireless Sensor Networks (WSNs) have potential as communication architecture applied to swarms.

A scenario for the communication both with and within the swarm has been proposed. The challenges implied by tough environmental conditions call for heterogeneous approach such as the one proposed in this dissertation. A communication within the swarm is held using short-range communication technology such as XBee-PRO modules. All nodes acting as sensing agents are equipped with short-range communication technology. The communication with the swarm is held using nodes acting as gateways to the shore equipped with long-range technology, such as SX1272 modules from Semtech, called LoRa. The deployment of nodes acting as gateways to the shore, called buoys, with fixed localization has been proposed. Each node in the network is aware of the GPS coordinates of buoys, thus in case of communication loss, it can orient itself in the direction of the nearest buoy, increasing chances of successful communication with the base station.

The short-range communication XBee-PRO technology has been tested in order to determine communication range with and without Line of Sight (LoS). The objective is to improve the range of the communication link, which nowadays in held via Wi-Fi in the distance around 30 m. The results were promising for real-world implementation into swarms of aquatic surface drones.

Keywords

RF communication, XBee, LoRa, Unmanned Surface Vehicles, USVs, multi-agent systems, swarms of aquatic drones, WSNs, short-range communication, long-range communication

Resumo

Os veículos não-tripulados de superfície aquática (VNTSA) têm um enorme potencial para uma vasta gama de atividades marítimas, tais como a monitorização ambiental ou a amortização da vida marinha. Estes veículos podem ser também aplicados em missões militares de apoio ao exército, em situações potencialmente perigosas, tais como missões de reconhecimento ou de vigilância. Os USVs (VNTSA) são capazes de realizar diversas tarefas devido ao progresso tecnológico e à redução do tamanho dos equipamentos nos últimos anos. A redução do tamanho e preço, acompanhado pelo aumento da fiabilidade, permitiu o desenvolvimento de sistemas multi-agente em larga escala.

Os sistemas multi-agente dos USVs autónomos aquáticos podem agir como um sistema de sensores distribuídos, melhorando o desempenho global, quando comparado com o desempenho de uma unidade de USV isolado, ou seja, quantos mais unidades no sistema, maior área monitorizada. Uma abordagem promissora, inspirada na natureza, é inteligência de cardume (swarm intelligence), que pode ser observada na população de insetos, como por exemplo nas formigas. O comportamento do cardume (swarm) é um movimento de um grande número de elementos, em que cada um é autónomo, porém só em grupo são capazes de resolver tarefas complexas. A troca de informações entre as unidades é essencial para o sucesso do grupo. A arquitectura de comunicação aplicada a cardume pode ser suportada pelas Redes de Sensores Sem Fios (RSSF).

Nesta dissertação, foi proposto um cenário para a comunicação "com" e "dentro" do cardume. Os desafios decorrentes das condições ambientais difíceis exigem uma abordagem heterogénea, tal como o proposto nesta dissertação. A comunicação dentro do cardume é realizada utilizando tecnologia de comunicação de curto alcance, tais como os módulos XBee-PRO. Todos os nós atuam como agentes de deteção, que estão equipados com tecnologia de comunicação de curto alcance. A comunicação com o cardume é concretizada usando os nós que funcionam como portas de comunicação (gateways) para a estação de base equipada com tecnologia de longo alcance, tais como os módulos SX1272 de Semtech, chamados LoRa. Foi ainda proposto, implementar nós, atuando como portas de comunicação (gateways) para a estação de base, designados de bóias, com localização fixa. Cada nó na rede tem conhecimento das coordenadas de GPS das bóias, assim, em caso de perda de comunicação, podem (re)orientar-se na direção da bóia mais próxima, aumentando a probabilidade de sucesso de comunicação com a estação de base.

A tecnologia de comunicação de curto alcance XBee-PRO foi testada, a fim de determinar o alcance da comunicação com linha de vista e sem linha de vista. O objetivo foi melhorar o alcance da ligação (link) de comunicação, que hoje em dia é realizado via Wi-Fi, a distâncias de aproximamente 30 m. Os resultados revelam potencial para a implementação no mundo real dos drones aquáticos de superfície.

Palavras-chave

Comunicação RF, XBee, Lora, veículos de superfície aquática não-tripulados, USVs, sistemas de multi-agente, cardumes de drones aquáticos, RSSFs, comunicações de curto alcance, comunicações de longo alcance

Contents

List of Figures

List of Tables

List of Acronyms

GPS	Global Positioning System
HANCAD	Heterogenous Ad-hoc Network for Coordination of Aquatic Drones
IEEE	Institute of Electrical and Electronics Engineers
IT	Instituto de Telecomunicações
MAC	Media Access Control layer
OSI	Open Systems Interconnection
PHY	Physical layer
PCB	Printed Circuit Board
UBI	Universidade da Beira Interior
USV	Unmanned Surface Vehicle
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network

Chapter 1

Introduction

1.1 Introduction

Maritime tasks are usually performed by human operated vessels, that makes them expensive in execution. They are low-resolution, both in time and space, due to the involvement of significant number of resources. Some of them are even dangerous, e.g. rescue or military missions like reconnaissance or surveillance. The concept of Unmanned Surface Vehicles (USV) supporting human in maritime activities is not recent, [9] presents an overview on USV´s worldwide development. In last years the radio communication industry has grown fueled by digital and RF circuits improvements, large-scale circuit integration, and other miniaturization technologies which make the equipment smaller, cheaper and more reliable [10]. That makes it feasible to consider large scale systems of autonomous USV.

Multi-agent systems consisting of autonomous drones have potential in mentioned activities, significantly decreasing operational costs and eliminating risks of accidents affecting human performers. They can be applied to variety of tasks such as sea-border patrolling, search and rescue missions, sea demining, marine oil spill clean-up, environmental monitoring, water movement tracking among others [11], see Figure 1.1.

Multi-agent aquatic systems can contribute in allerting about natural disasters as well as monitoring sea-life. There are various institutions, which might be interested in harvesting data from the swarm such as temperature, amount of plankton, presence of some species or even radiation.

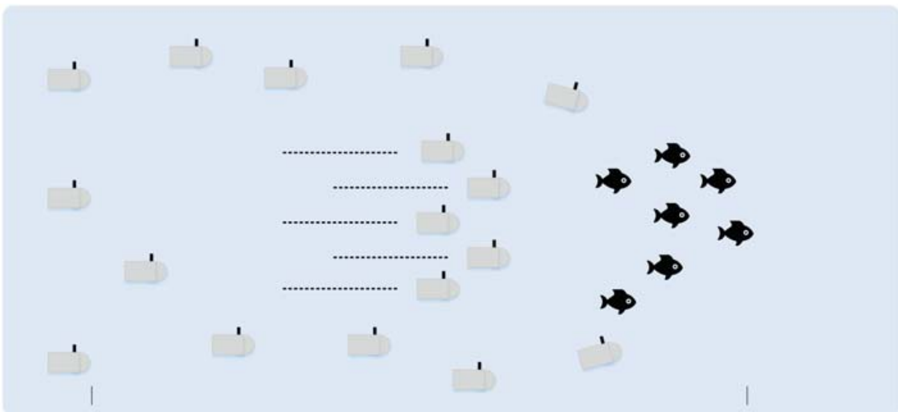


Figure 1.1: One of applications of a swarm of aquatic drones is fish following.

A significant advantage of distributed sensing applied to multi-agent system is spreading tasks over multiple agents such that failure of one does not affect the global scenario [12]. In order to keep the solution low-cost and scalable, each agent must be relatively simple and inexpensive. Autonomous systems composed of many of such agents are called swarms.

Swarm robotics is an approach inspired from self-organized behaviour found in nature. For instance, there are many examples of swarm intelligence in social insects and slime moulds [13]. Individual unit is cognitively relatively simple and restricted in what it can achieve, whereas the group collectively is capable of wonderful achievements.

The exchange of information between units is essential for the success of groups´ mission. Indirect communication via the environment, works well for insects, but it seems clear that it is not always the most ideal communication channel for information exchange in aquatic environment [14]. Radio communication is a good medium, which enables local interactions. The links between units need to be preserved in order to maintain the data flow into swarm of aquatic drones.

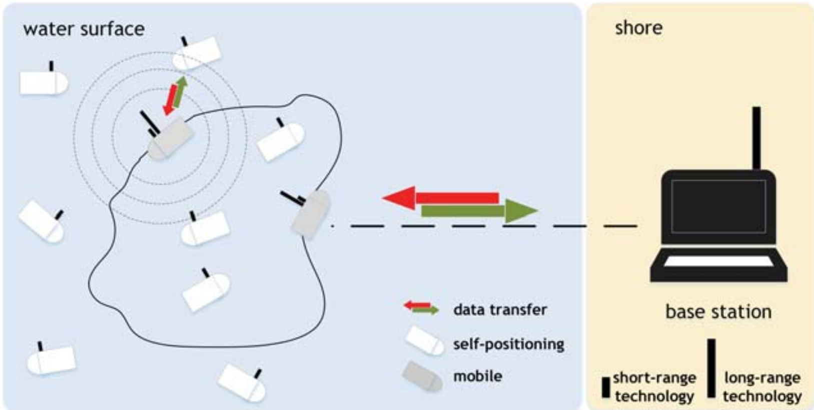


Figure 1.2: A swarm of drones is a decentralized system, where agents act based on local information to accomplish the global goal.

Effective communication both with and within a swarm of aquatic drones carries various challenges, due to high humidity, waves and presence of two mediums. Moreover, different types of missions have different requirements. These issues, combined with the need of dynamic ad-hoc network technologies, call for heterogenous communication approach, see Figure 1.2.

1.2 Objective

The aim of this work is to develop a network architecture which allows communication both with and within a swarm of aquatic drones. The exchange of data is a high priority issue, because more information the drone receives, more it knows about the environment, thus its performance is better.

In detailed description, principal objectives if this thesis are:

- To develop the communication system prototype, with both short and long-range communication capabilities;
- To develop a heterogeneous communication scheme in order to maintain local and global interactions;
- To propose communication technologies adequate to Heterogeneous Ad-Hoc Network for Coordination of Aquatic Drones (HANCAD) project.

More specified objectives, which have been realized in this work are:

- The development of printed circuit suitable for two technologies of both short and long-range capabilities;
- The development of communication protocol in C++ language;
- The evaluation of proposed solution through and indoor and outdoor experiment in laboratory of Instituto de Telecomunicações and nearby aquatic environment respectively.

1.3 Contributions

This research introduces heterogeneity into hierarchical network based on IEEE 802.15.4 standard. In order to define communication range of the radios and importance of the Line of Sight (LoS), both above the ground and above the water surface, as well as the trade-offs associated with the multi-hop communication, there have been performed tests.

By far, this work has been appreciated by two accepted publications and presented internationally. First one was published in 82nd Vehicular Technology Conference: VTC2015-Fall, IEEE, in Boston, USA, September 2015. It discusses wireless sensor network technologies for swarms of inexpensive aquatic surface drones in the context of HANCAD project. The second paper was published and presented in 10th Conference on Telecommunications – CONFTELE, in Aveiro, September 2015. It reported on preliminary fields trials conducted to validate the potential of the XBee-PRO modules, in terms of coverage, range and multi-hop capabilities, in maritime scenarios.

The results will support research in behavior of swarm of aquatic drones in real-world conditions. It is part of Heterogeneous Ad-hoc Network for Swarms of Aquatic Drones (HANCAD) project, run in Instituto de Telecomunicações (IT) IT-DEM, Universidade da Beira Interior, Covilhã, IT-Aveiro, University of Aveiro and BioMachines Lab, ISCTE, Lisbon. It is planned to perform simulations in order to develop and evaluate controller for an aquatic surface drone and test the solution in real-world environment.

The experiments on packet delivery performance in laboratory conditions have been performed in order to determine the importance of time interval between transmission. Results are satisfing the requirements of HANCAD project. Besides, an experiment on routing performance in multi-hop communication with one intermediate node has also been performed. The broadcast scenario both in single and multi-hop scenario had high performance, while unicast communication in multi-hop transmission showed the impact of routing discovery process on efficiency of mesh protocol.

1.4 Challenges

The proposed swarm of aquatic drones involves the development of communication network using industrial, scientific and medical (ISM) radio bands, which are unlicensed shared bands. Long-range communications, in turn, such as 3G/GPRS or satellite, usually implies high cost of necessary hardware and operation.

The network would consist of energy-efficient nodes (drones) and collaborative base stations, which may issue some command to the swarm. Base stations preferably use renewable energy sources (wind, solar, etc.).

Swarms would typically be dispersed over a large area (tens or hundreds of square kilometres), and a base station should be able to issue high-level instructions, and to receive data from the swarm. In terms of exchange of data, all nodes need communication access to others to provide data transfer between units.

Communication architecture should be heterogeneous, as long-range communication usually increased energy consumption and delays, only few drones may be equipped with those modules. In this way, drones with long-range communication capabilities serve as gateway to the base stations.

As the communication is essential for a success of a swarm, the loss of one agent can not influence the others.

1.5 Structure of the thesis

This dissertation is divided into five chapters, including this one, and two appendices. It presents studied matter and conclusions, which enabled preparation of the network's structure appropriate for the swarm of aquatic surface drones.

Chapter 2 presents the State of the Art. It introduces the term of Wireless Sensor Networks (WSNs) and provides principals of WSNs. It addresses the OSI model applied to WSN, MAC protocols, routing protocols and topology control.

Chapter 3 presents the heterogenous communication technologies for swarm of aquatic drones. It presents concept of the project and proposed solution.

Chapter 4 presents implementation process and results of tests. It evaluates the proposed solution as well.

Chapter 5 presents the final evaluation of the project and concerns future work.

Appendix A contains pinouts of communication modules such as XBee-PRO and LoRa, pinout of Raspberry Pi, which were used to develop integrated hardware solution for aquatic drone.

Appendix B presents the Drone Remote Console used for monitoring and control of swarm of aquatic drones in HANCAD project.

Chapter 2

State of the art

2.1 Introduction

Effective communication both with and within a swarm of surface drones presents a variety of challenges, due to the extent of environment, presence of waves, and potential signal reflections. The rapid progress in technology in recent years has led to a number of different approaches, such as Wireless Sensor Networks (WSNs).

Wireless Sensor Networks are distributed sensing systems designed to monitor physical conditions, such as temperature, humidity, etc, and pass information through the network to the main station, see Figure 2.1.

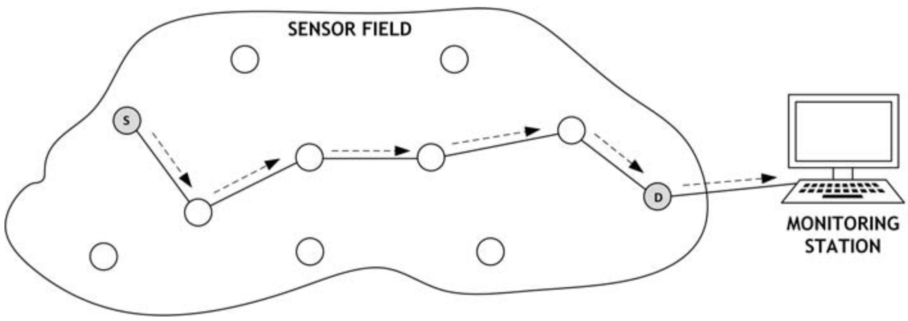


Figure 2.1: WSN network.

Deployment of WSN in ocean environment is not easy due to meteorological conditions, waves, strong water currents, tides or surrounding terrain (e.g. ice bergs, rocks, cliffs). Since the network is located on the water surface, nodes must be highly resistant to water as well as to sun. Each node must be robust to physical damages caused by collisions with obstacles. Static nodes require an anchor or self-positioning mechanism.

Usually WSN nodes share a common communication medium. It is much easier to keep communication over smooth water surface, waves on the sea surface cause node´s movement along vertical axis. Antenna´s oscillations make difficult to keep line-of-sight (LoS), which may cause propagation losses. Electromagnetic waves are strongly absorbed under water surface, hence the signal is lost in long distances. Acoustic communication is an alternative, but it greatly affects a cost of each node and causes big delays.

As the nodes are often difficult to access after deployment, they need to be energy-efficient. Reduced energy consumption of all sensing, communication and power module is desired. As batteries have limited energy, ideally each node should have renewable energy source (e.g. solar panel) or energy harvesting device which recharges the batteries on board when possible. Research in harvesting energy problem should advance, due to the contamination problem from traditional batteries.

To maximize the scalability of aquatic WSN, each node must be inexpensive. The objective is to develop a solution 'low-cost', while improving resolution of the activity both in time and space.

Size of WSN can vary as well as a topology, according to the purpose it serves [15]. It can be stated that communication range of the network is proportional to its size, depending on the implemented communication protocol.

2.2 OSI model

In order to investigate the topic of MAC or routing protocols there must be introduced Open Systems Interconnection (OSI) model first.

The OSI reference model, according to CISCO, describes how information from a software application in one computer moves through a network medium to software application in another computer [16]. It is considered as the primary architectural model for intercomputer communications. All tasks involved with information flow between computers are divided into seven manageable groups, which are assigned to each of seven OSI layers. The highest layer, the application layer, is the closest to the user. The lowest one, physical layer, is the closest to physical network medium.

The OSI model, see Figure 2.2, is not a method itself, it is a concept. It enables relatively easy one layer update without affecting on the other layers.

The information exchange process occurs between OSI layers. Each layer in the source system adds control information to data, so that the frame grows in size, and each layer in the destination system analyzes and removes the control information from that data. At the physical layer, the frame is placed onto the network medium. This process is called ecapsulation.
Description of OSI layers, according to CISCO:

1) Physical layer
   The physical layer takes care of transmission and reception of raw bits over a physical medium. It defines the electrical, mechanical, procedural, and functional specifications for activating, maintaining, and deactivating the physical link between communicating network systems, e.g., definition of characteristics such as voltage levels, physical connectors.

2) Data link layer
   The data link layer provides transit of data across a physical network link, data layer specifications define network and protocol characteristics, including physical addressing, network topology, error notification, sequencing frames. A data link layer address uniquely identifies each physical connection of a network device. The Institute of Electrical and Electronics Engineers (IEEE) has divided into two sublayers: Logical Link Control (LLC) and Media Access Control (MAC).

3) Network layer
   The network layer manages and defines structure of a network providing addressing and

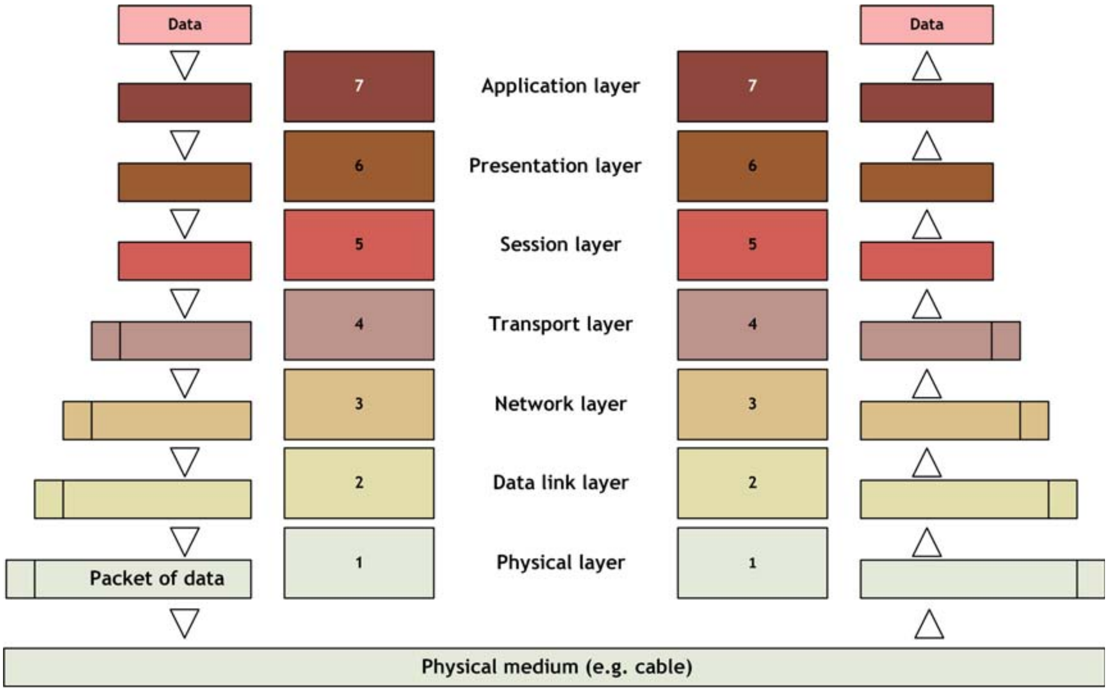


Figure 2.2: OSI model.

routing. On this plane, nodes are connected with each other within a network, each one identified by its network address.

4) Transport layer
The transport layer provides inter-network data transport services such as flow control, error checking. It creates packets out of the message received from the application layer.

5) Session layer
The session layer establishes, manages and terminates communication between computers. Communication sessions consist of service request and service responses that occur between applications located in different network devices.

6) Presentation layer
The presentation layer translates data between a networking service and an application, ensuring that information sent from application layer of one system will be readable by the application layer of another system. It provides a variety of coding and conversion functions for application layer data treatment such as character encoding, encryption/decryption and data compression as well.

7) Application layer
The application layer is the closest to the end user, which means that both the OSI application layer and the user interact directly with the software application. Its functions typically include identifying communication partners, determining resource availability and synchronizing the communication, identifying any constraints on data syntax.

2.3 MAC sublayer protocols

As WSN share the same communication medium, it needs to be managed efficiently. The Media Access Control layer is a sublayer of Data link layer of OSI model, see Figure 2.2, which coordinates access to the medium. The most power consuming element of WSN node is radio, which is controlled by MAC protocol. Efficient MAC protocol improves the throughput and profits in power consumption.

The MAC protocols in WSN has been widely researched, therefore it is necessary to investigate previous studies, while developing new protocols. Authors from [17] present evolution of WSN MAC protocols in recent years, pointing out a clue about why a protocol was proposed for a problem and what remaining issues lead to another solution, identifying the development flow.

Authors from [18] prepared comprehensive state-of-the-art study thoroughly exposing the prime focus of WSN MAC protocols and design guidelines that inspired these protocols. They prepared very useful thematic division classifying protocols according to the faced problem.

The design of MAC protocols for WSN must concern many factors such as high energy constraint, simplicity of nodes, their low computational capabilities as well as poor synchronization capabilities, thus MAC should be functional.

According to [18] there are two main approaches for managing access to a medium: contention-based and reservation-based approaches. Thus, most of MAC protocols is combination of above.

1) Reservation-based protocols:

Within reservation-based protocols knowledge of network topology is required to establish a schedule that allows nodes to access the channel and communicate with others. Time Division Multiple Access (TDMA) protocol is a good example of reservation-based protocol.

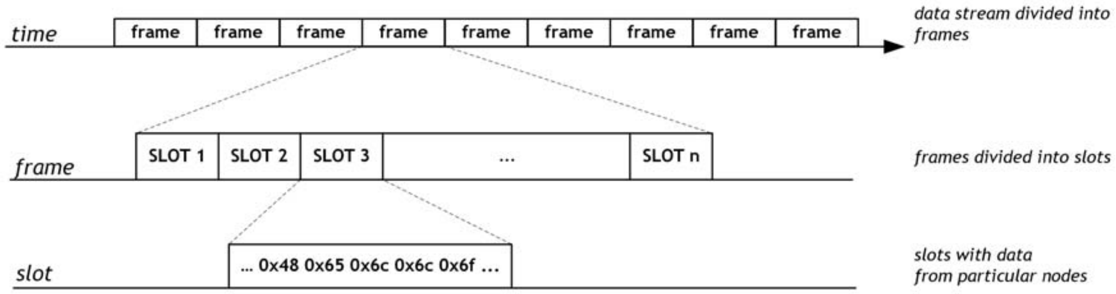


Figure 2.3: Framing in the TDMA protocol.

In TDMA, see Figure 2.3, time is divided into frames and each frame into slots. The nodes transmit one after another, each one using its own unique time slot. The schedule reduce collisions between interfering nodes, by avoiding their transmission at the same time and guarantees finite and predictable delays. As each node is assigned to unique slot, fairness among nodes is ensured.

2) Contention-based protocols:

Within contention-based protocols neither knowledge of network topology nor global synchronization is required. Nodes are allowed to transmit when the channel is clear (when no other node is transmitting at the moment). Canonical representative schemes of content-based protocols are ALOHA and CSMA (Carrier Sense Multiple Access).

ALOHA was pioneering system developed in 70s of last century in the University of Hawaii. The characteristics feature of ALOHA is that the nodes can start transmission at any time and that they do not listen the channel before. Each node is allowed to send its data to the hub, but occasionally collisions occur. There were proposed acknowledgement (ACK) and retransmission schemes.

In CSMA, a node having a packet to transmit first senses the channel. If the channel is busy, it delays its transmission in order to avoid collision. If the channel is clear, it starts to transmit the packet.
Contention-based protocols need more attention to collisions avoidance as opposed to reservation-based ones, where it is achieved by scheduling and slots assigning.

CSMA/CA (CSMA/Collisions Avoidance) is a technique widely used to reduce collisions. It introduces very small RTS (Ready To Send) and CTS (Clear To Send), which are exchanged between communicating nodes prior to the "right" transmission. The transmitter sends RTS to the receiver. When the receiver receives a RTS, it responds with a CTS, in order to reserve the channel around it and waits for a data packet. After the transmission the receiving node sends an ACK packet. If the ACK is not received, the packet is considered lost and a retransmission is arranged. There is no RTS, CTS or ACK packets exchange in broadcast transmission, they can be used only in unicast mode.

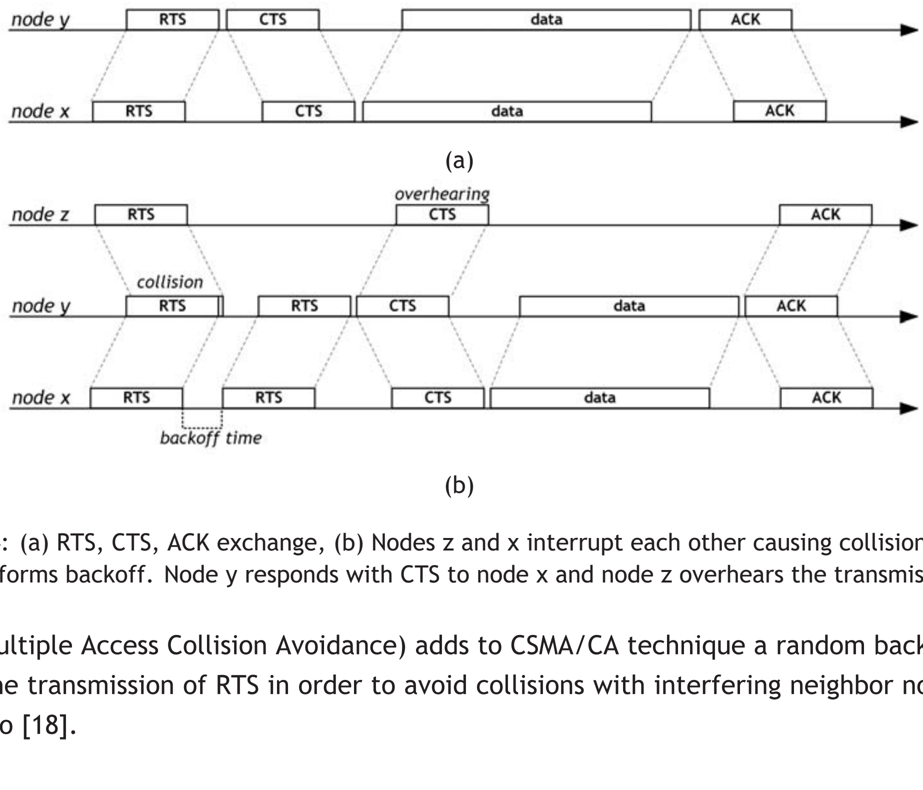


Figure 2.4: (a) RTS, CTS, ACK exchange, (b) Nodes z and x interrupt each other causing collision, so node x performs backoff. Node y responds with CTS to node x and node z overhears the transmission.

MACA (Multiple Access Collision Avoidance) adds to CSMA/CA technique a random backoff time before the transmission of RTS in order to avoid collisions with interfering neighbor nodes, according to [18].

3) Summary of reservation- and contention-based protocols:

Both reservation- and contention-based protocols have their advantages, their comparison is shown in Table 2.1. Authors from [18] did comparative study of both protocols, and stated that reservation-based protocols outperform the contention-based ones in terms of throughput, fairness and delays. However it is achieved at large cost of global synchronization and loss of

Table 2.1: Comparison of the characteristics between reservation- and contention-based protocols.

	Contention-based	Reservation-based
Fairness	Nodes compete with each other for the access to the channel.	Each node has a unique slot in the frame assigned
Mobility	They do not require knowledge of network topology, so the network may be dynamic	As slots are assigned to nodes, there is no possibility to vary.
Delay	They may suffer from delays due to collisions with other interfering nodes.	Communication is scheduled, so delays are finite and predicted.
Throughput	When the traffic-load increases they suffer in performance.	In highly loaded networks scheduling increases the throughput, but it is limited and it can be increased beyond the utilization of all available slots.
Cost	There is no need for central hub, managing the scheduling and synchronization.	Both knowledge of topology and synchronization require large overheads and synchronization.

mobility, which is crucial element in HANCAD project.

2.4 IEEE 802.15.4 and ZigBee specification in OSI model

IEEE 802.15.4 is a standard created, developed and maintained by IEEE 802.15 working group of the Institute of Electrical and Electronics Engineers (IEEE). It was developed in order to satisfy requirements of low-rate wireless personal are networks (WPAN). It specifies both physical and MAC layer focusing on low-cost, low complexity and low-power consumption. It operates in one of three unlicensed frequency bands: 868 MHz band ranging from 868.0 to 868.6 MHz in Europe allowing one communication channel; 915 MHz band ranging from 902 to 928 MHz in North America allowing up to ten channels; and 2.4GHz band between 2400 and 2483.5MHz, which is used worldwide with sixteen channels available. The IEEE 802.15.4 technology enables connection of devices using CSMA/CA channel access technique, guaranteed time slots for communication (GTS) and includes some power management functions such as radio transceiver activation or desactivation (duty cycle), link quality evaluation or energy detection. It defines mechanisms of forming and joining the network. The physical layer is capable of clear channel assessment and channel selection. Moreover, it established RF link between two devices, modulates and demodulates bits, synchronizes devices, transmits and receives packets, as presented in [19].

IEEE 802.15.4 standard organises transmission into frames, which can differ depending on the purpose, e.g., data frame, ACK frame, among others, and support two network topologies: Star and Peer-To-Peer, often called Mesh, see subsection 3.2.2. However it does not specify address assignment in multi-hop communication. The MAC sublayer performs security control, optional star topology functions, such as time slot management or generation of ACK frames.

IEEE 802.15.4 specifies only PHY and MAC layers, thus there are other options to implement as upper layers such as ZigBee specification, see Figure 2.5.

Network layer in ZigBee network provides routing and establishes topology of the network (star, mesh or cluster tree). It performs network initialization, assigns node addresses, configures and discovers new nodes, performs the discovery of other networks and provides security.
ZigBee Application Support Sublayer (APS) filters out packets, which are not destined to the

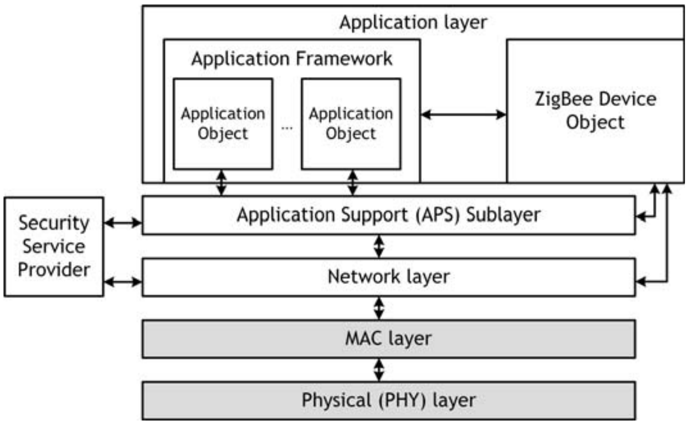


Figure 2.5: ZigBee specification on OSI model.

node, informing the sender whether the packet was received or not by generating end-to-end acknowledgments, performs automatic retries, if ACK is requested, maintains various tables from application-level such as binding table (methods about connecting one node to another) or address maps. Application Framework is a set of routins, or API, that have been developed to interact with ZigBee radio. Application Framework and APS form together the ZigBee interface, see [20].

ZigBee Device Object is an application running in every End-Point device. It provides an interface to network layer management services, such as discovering, joining or leaving the network. ZigBee Device Object provides Zigbee Device Profile, through which client-server services can be called. Client services are requests, server services are responses.

2.5 Routing protocols

Routing is the process of selecting best paths in the network. Researchers have proposed already hundreds of routing protocols. The goals of these protocols are similar: maximize throughput, while minimizing packet loss, control overhead and energy usage. However, the priorities vary depending of the application.
Early protocols had been classified into on-demand and table-driven protocols, but between these two, several hybrid approaches have been developed. Authors of [21] compared some examples of protocols from each class. Below definitions may quote above paper.

Source-initiated routing (called also on-demand) represents a class of protocols where the route to a destination is created on demand of the source. Route discovery is a process when the route to a destination is defined through flooding the network with route request packets, starting with the nearby neighbors of the source. Route discovery process ends when one or more path to a destination are found. An example of source-initiated protocol is Ad Hoc On-demand Distance Vector (AODV).

A source, which wants to send a data packet to a destination node, must check before the transmission its routing table for a valid path to destination node. If a route exists, it forwards the packets to the neighbor saved in the routing table. If the route does not exist in the table, the source must start a route discovery process. It broadcasts a route request (RREQ) to its neighbors, packet is forwarded further until RREQ reaches the node with the route to the destination or the destination node itself. The frame contains source node address, destination node address, last known number of hops, and current sequence of hop. When the RREQ travels through the network, each intermediate node adds a record to its own routing table with the address from the neighbor from which the first copy of packet arrived. If the same RREQ packet is received later, it is discarded. This information is used later in path-reconstruction process called reverse path for route reply (RREP) packet. Figure 2.6 presents route discovery process and reverse path formation. RREP packets are forwarded in the network along the established reversed path. If a route is broken, maintenance process will be performed by sending a link failure notification message to each of neighbors, in order to delete that particular path. When this message reaches the source, it re-initiates the route discovery process.

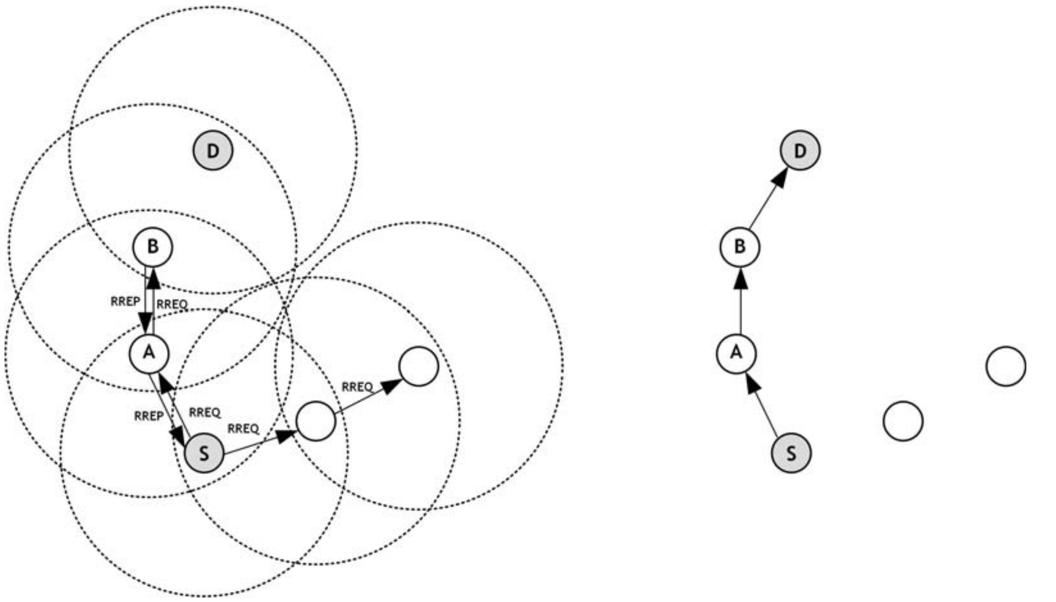


Figure 2.6: Discovery process in AODV protocol.

Table driven-protocols maintain up-to-date routing tables with routes from each node to every other node in the network, in order to reduce route maintenance overhead. Route updates are propagated through the network keeping the information as recent as possible, thus these protocols are suitable for highly dynamic networks.

Location-aware protocols assume that all individual nodes have knowledge of localization (e.g. Global Positioning System - GPS) of all other nodes in the network. Authors from [22] describe Geographic Random Forwarding (FeRaF) protocol, which uses geographic routing. When a source node has data to send, it broadcasts a RTS packet to neighbors. When neighbors receive RTS, they assess their own priority in trying to act as an relay, based on how close they are to the destination. The coverage area faced to the destination is called forwarding area and it is split into *N* regions with different priorities based on their distance from destination. Higher priority, smallest distance to destination. The source waits for CTS from the node in highest priority region. If it does not receive any, it means that the region is empty and the source broadcasts the message again polling the neighbors with lower priorities. If it receives one, the relay node is selected, see Figure 2.7. If it receives more than one, the neighbor closest to destination is chosen as relay. The relay node forwards the message, keeping original source location and destination location, thereby providing geographical route without any routing tables or network topology information. This protocol is suitable for stationary nodes, densely deployed, and randomly turn on and off, providing a random topology. According to authors, if the density of active nodes is appropriate, it is likely that the node closest to destination will provide an advancement towards the destination. However there is no guarantee that a source will always be able to forward the packet towards the destination.

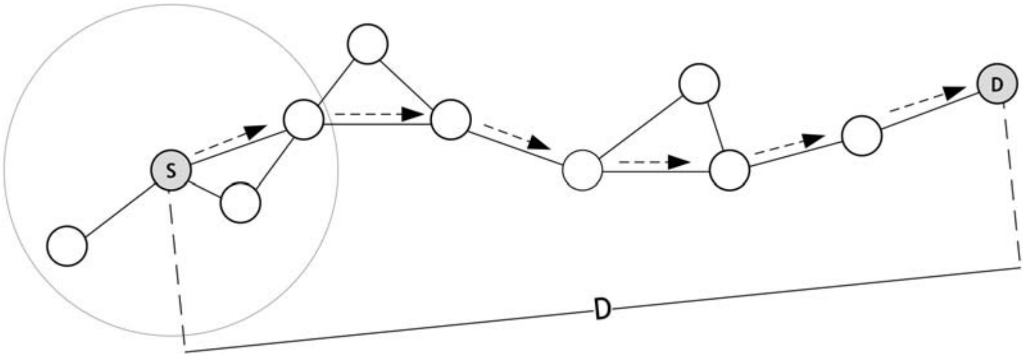


Figure 2.7: Location-aware protocol.

Multipath protocols create multiple routes from the source to destination as opposed to conventional protocols. This solution improves the delivery reliability and helps network is overload, which may happen due to bursty traffic within a network. Routes usually expire due to node mobility, thus they are generated on-demand. Ad Hoc On-demand Multipath Distance Vector routing uses the basic AODV scheme, with extensions to create multiple loop-free paths. In these protocols it is assumed that each RREQ and RREP messages arriving to the intermediate node might potentially use different paths, but all of them can not be accepted, because they can cause creation of loops, see Figure 2.8. A maximum number of hops value was introduced, which is compared to the number of hops counted for paths recorded in nodes. A path with larger number of hops than the maximum is discarded since probably it has a loop.

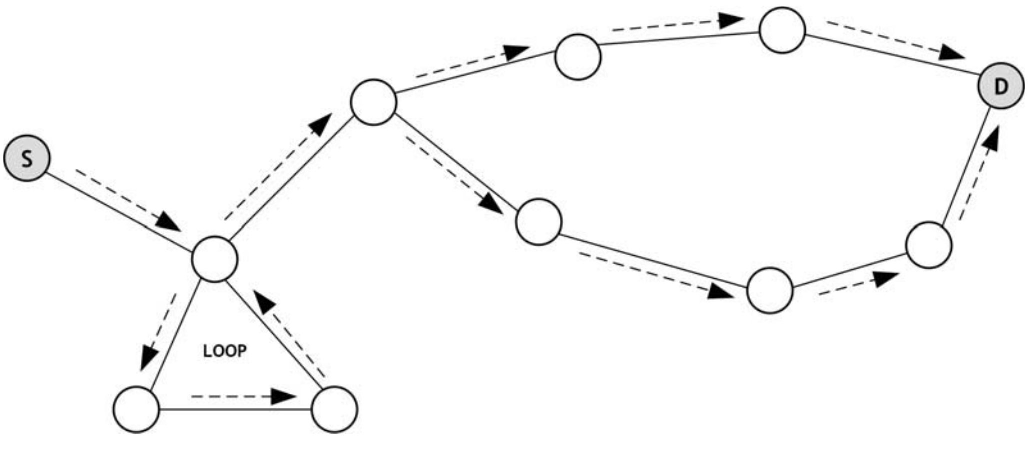


Figure 2.8: Multipath protocol.

In hierarchical routing a hierarchy of nodes is built, typically through clustering, see Figure 2.9. Higher level nodes provide special services, improving efficiency of routing. The goal of Hierarchical state routing (HSR) is to replace flooding of control information. It introduces local collection of information in the clusterhead, further forwarded to other clusterheads. Basically HSR introduces a set of protocols based on multi-level clustering. Nodes from the lowest level elect a local clusterhead, which will act as router. When a source wants to send a packet, it sends it to local clusterhead, which checks the destination and forwards it to the nearest gateway node. The gateway node forwards it to another node which will deliver it to one higher in hierarchy. The process continues until the packet reaches the gateway of destination cluster. The final gateway sends received packet to the clusterhead of destination cluster, which delivers it further to destination node.

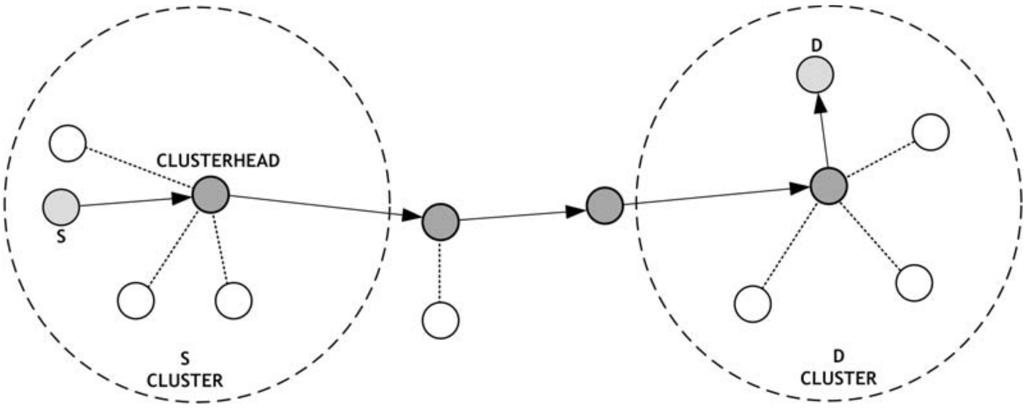


Figure 2.9: Hierarchical routing protocol.

2.6 Topology control

Topology control is a technique used in WSNs to improve energy efficiency by management of certain node parameters such as transmission power or switching modes of operation while maintaining network connectivity and sensing coverage. According to [1] topology control function must be a software layer located between the data link control layer and the network layer, interacting with both.

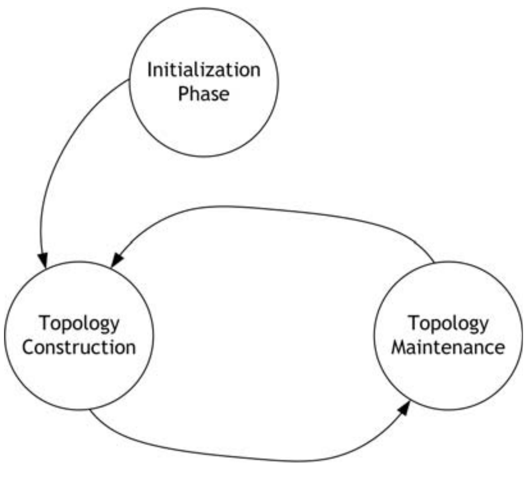
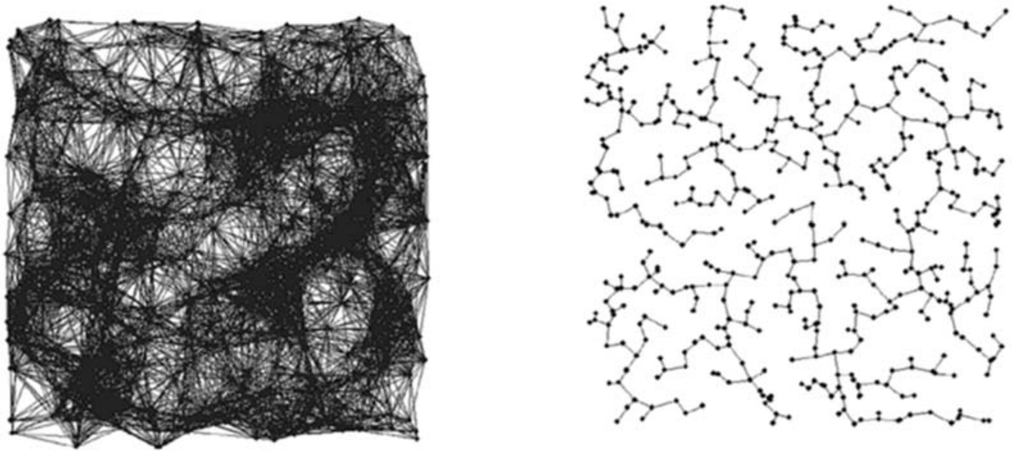


Figure 2.10: Topology control process.

Topology control is an iterative process, see Figure 2.10. First, nodes build the initial topology by using the maximum transmission power discovering all neighbors within their range. The following phase is Topology Construction, when new reduced topology is built.

Figure 2.11 presents an example network of nodes, which transmit with their maximum transmitting power compared to the same network after the topology construction mechanism.



(a) The Maximum Power Graph

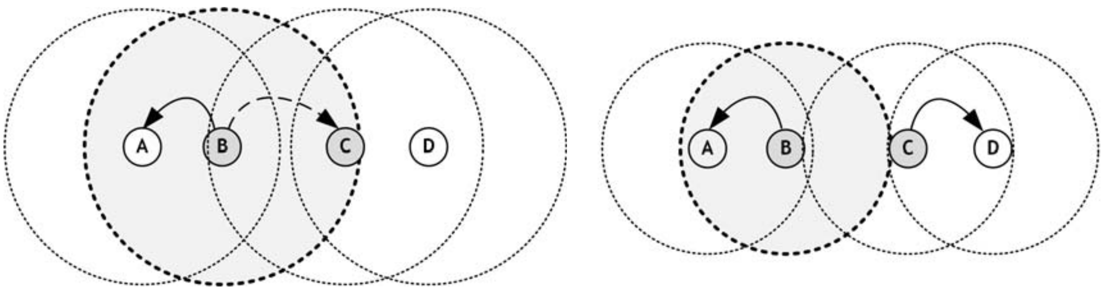
(b) The reduced topology after the topology construction mechanism

Figure 2.11: Comparison of topologies (images from [1]).

As soon as this phase establishes the reduced network, the Topology Maintenance phase starts. It is an algorithm monitoring the status of the reduced topology and executes a new topology construction phase when needed. This cycle may be repeated many times over the lifetime of the network.

Topology construction has impact on energy saving and preserving network characteristics such as connectivity and coverage. Decreasing the transmission power of the nodes eliminates certain direct links and forces packets to go through multiple hops. Moreover, nodes can switch operation modes to ON/OFF when they are unnecessary in order to spend their energy fairly and prolong the network lifetime.

The topology construction phase affects as well performance on data-link layer by reducing packet collisions and number of retransmissions, thus decreasing communication cost, which might be an issue in densely deployed network.



(a) Node C is overhearing.

(b) Transmission power is reduced.

Figure 2.12: The exposed terminal problem.

Reducing the transmission power can eliminate the exposed terminal problem. For example, in Figure 2.12a node B is transmitting to node A and node C overhears the transmission, thus node C can not transmit its data to node D. Both node B and C could transmit their data simultaneously to respectively node A and D, if their transmission power was reduced, see Figure 2.12b.

Many events might occure in the network, changing network topology with time, thus topology maintenance should be applied.

2.6.1 Topology control in communication stack

After the nodes have been deployed, data link layer (DLL) starts node discovery with maximum transmission power, see Figure 2.11(a). Next, it communicates to topology control layer to start topology construction phase. During this phase, according to implemented algorithm, topology control layer changes the transmission power and manages modes of nodes. Once, the topology is reduced it signalizes to network layer to start route discovery process. After that, topology maintenance phase is initiated in order to monitor events in the network and to inform if topology needs to be changed (locally or globally). If the topology needs to be changed, data link layer must perform changes in topology, then network layer executes route discovery. The process runs always in this order. Topology construction mechanism might be also triggered when topology control layer receives the notification from network layer that a route is no longer available. The interactions between layers are significant, especially when network contains mobile nodes.

There have been proposed various schemes of topology construction and topology maintenance. Authors from [23] state that topology control in wireless sensor networks is a growing field, but unfortunately most proposals have no real-world implementation, because a large part of approaches is focused on theoretical assumptions. However integration of topology control with other OSI layers is a challenging task.

2.7 Wireless Sensor Network architecture

2.7.1 Node architecture

Nodes usually consist of four modules: (a) power supply module, (b) communication module, (c) processing module, and (d) sensing module, see Figure 2.13. The most common power source for a WSN node is battery, which can be replaced or recharged [24]. Larger network size is, more problematic and less economical is battery replacement, hence the energy consumption of a sensor node should be controlled. Energy exists in the environment in many kinds (e.g. solar energy, thermal energy, wind energy and wireless energy among others), hence power supply module may consist also of energy harvesting device and power management system. Communication module is a wireless communication transceiver (RF transceiver) with antenna used to send and receive data to and from other nodes. Processing module is usually a microcontroller able to process information collected by sensors and pass it to the communication module, so it may be forwarded to another nodes. The content of sensing module may vary depending on the application of WSN. These may different sensors such as temperature, humidity, pH and others.

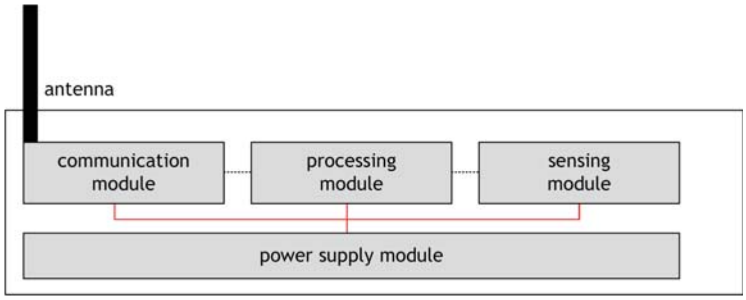


Figure 2.13: WSN node architecture.

Concurrent execution of several processes, with communication among them, is crucial for WSN nodes [2]. They have to handle data incoming from various nodes in arbitrary intervals of time. Simple non-concurrent model has a risk of loosing sensor data, while received packet is processed or loosing a packet, while sensor information is processed. The node should have reactive nature, waiting for an event to happen, while an event might be an incoming packet or sensor information. This approach is called event-based programming, see Figure 2.14.

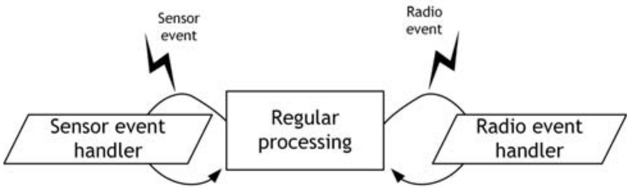


Figure 2.14: Event based programming model (adapted from [2]).

2.7.2 Network architecture

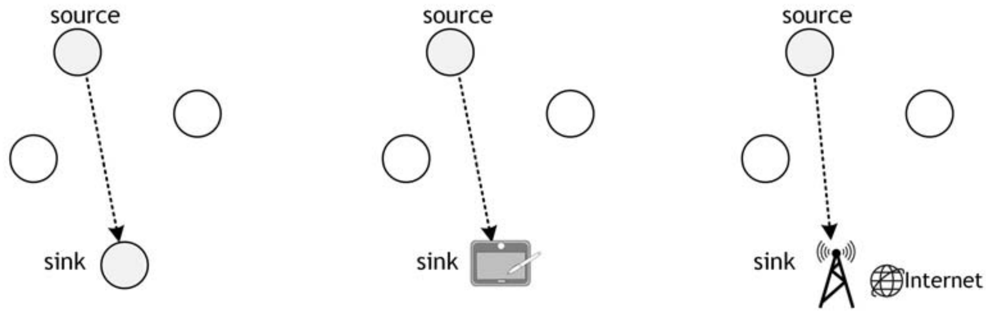
WSN consists of sources and sinks. A source is every node, typically a sensor node, which provides a piece of information such as temperature measurement. A sink is an unit, which gathers the information from the network. A sink might be (i) one of the nodes within the network, (ii) a device used to interact with WSN, or (iii) a gateway to another network, e.g., Internet, [2]. Figure 2.15 presents different types of sinks.

Due to radio power concerns the direkt link between a source and a sink is not always possible, especially in WSNs, which from assumption spread over large areas. Multi hopping mechanism has been considered in order to overcome these obstacles. In fact, multi-hop solution is more energy-efficient in range of distances than single hop one.

2.8 Summary and conlcusions

Effective communication both with and within a swarm of surface drones presents a variety of challenges due to humid and unstable environment, limitations in design such as size, power consumption or tough weather conditions such as strong wind or fog, thus heterogenous approach is desirable.

Wireless sensor networking is promising approach for distributed sensing system such as swarm of aquatic drones. Low network layers such as physical layer or MAC sublayer are specified by



(a) Sink as one of nodes within WSN. (b) Sink as a device from outside the network. (c) Sink as the gateway to another network.

Figure 2.15: Types of sinks.

IEEE 802.15.4 standard for wireless personal area network (WPAN). IEEE 802.15.4 standard have been briefly described in this chapter and ZigBee specification, which is a base of XBee-PRO radios, has been presented in the description of OSI architecture.

MAC protocols has been widely studied for years. They might be divided into three groups: reservation-based, contention-based and hybrid. CSMA/CA technique is an example of contention-based protocol, often implemented not only in IEEE 802.15.4 standard-based networks but also in 802.11g standard-based commonly called Wi-Fi.

Hundreds of routing protocols have been presented already. Routing process is very challenging in a swarm due to high mobility of drones. The most suitable ones seem to be on-demand routing and location-aware protocols.

The main objective of topology control is to improve energy efficiency of the network by switching ON and OFF the nodes and transmission power adjustment. In this investigation no schemes of operation mode switching or transmission power management are proposed, because power consumption is not a first priority. However a scenario has been proposed for communication with and within a swarm of aquatic drones.

Chapter 3

Heterogenous communication technologies for swarms of aquatic surface drones

3.1 Introduction

The key challenge for the communication among nodes within the swarm of aquatic surface drones is not only to maintain connectivity between agents of the swarm, but also to perform mission's tasks, which could limit agents motion, [25], e.g., during the agent must inform the base station about the intruder during surveillance mission, and it must not loose the intruder from sight. The mesh networking and dynamic routing is promising for maintenance of the link to the base station. From the assumption drones must be simple and low-cost in order to keep the multi-agent system approach feasible. As all drones execute goals of the mission, only some of them might have long-range communication capabilities.

3.2 Communication solutions

There is a variety of communication technologies available on the market, e.g. Bluetooth, WiFi among others. From assumption communication between agents in swarm should be cheap, because in general operation within swarm should reduce operational costs of maritime activities. This condition eliminates GSM standard. The objective of this research was to extend the communication range of agents and to provide long-range communication to the control station. Before drones in the developed swarm communicate through an IEEE 802.11g (Wi-Fi) ad-hoc wireless network by broadcasting UDP datagrams, see Figure 3.1. The approximate obtained maximum range was 30 m. The processing unit of the robot is Raspberry Pi 2.

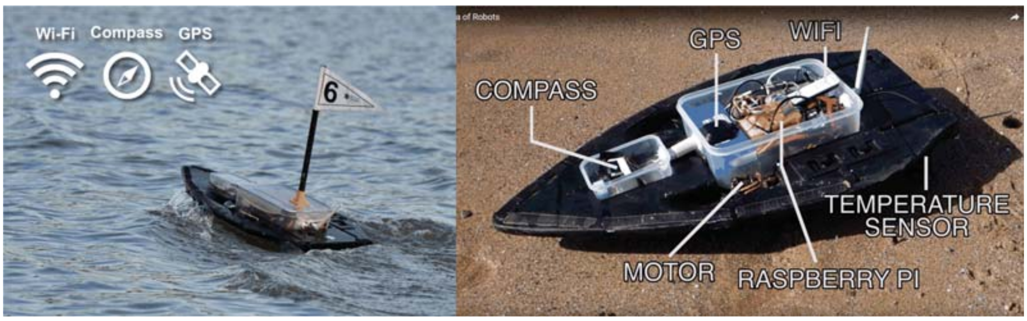


Figure 3.1: Robot scheme (images from [3]).

The goal is to replace Wi-Fi with XBee-PRO DigiMesh 2.4 GHz from Digi International Inc., extending the communication range up to 1.5 km (according to the datasheet) in Line of Sight (LoS). As the long-range technology we propose LoRa from Semtech Inc. LoRa (SX1272 module) operates at 860 MHz and is a low power long range transceiver that providing communication range up to 22 km, as claimed in [5], and high interference immunity while minimizing current consumption.

3.2.1 Waspmote

The primary idea was upgrading the communication module by adding Waspmote platform. It is an open-source solution developed by Libelium, designed for WSNs, with low energy consumption modes. The great advantage is its modular architecture and compatibility with the Arduino programming environment (IDE). It is an Atmega microcontroller based device (ATmega 1281) powered from batteries with temperature sensor and accelerometer built-in, measurements data can be saved in micro-SD card, see Figure 3.2. Waspmote allows various technologies for communication: 802.15.4/ZigBee standard, LoRa, Wi-Fi, 3G, GSM/GPRS among others.

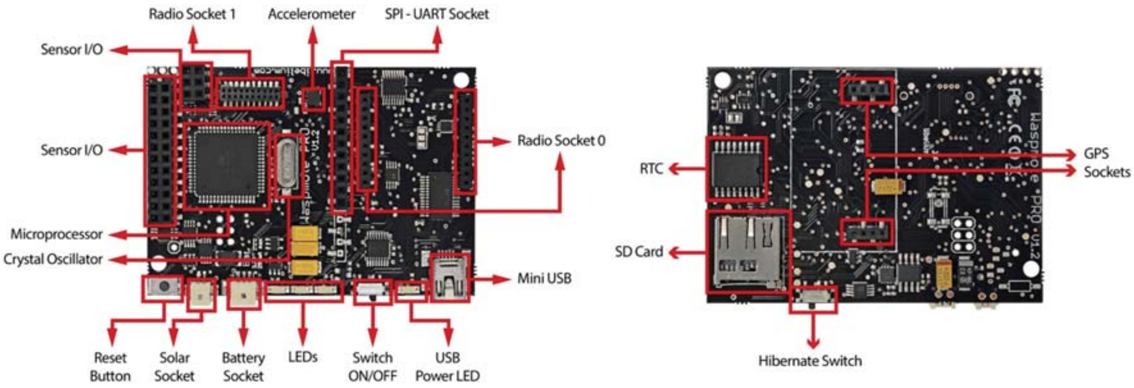


Figure 3.2: Waspmote platform (i) up view, (ii) bottom view (images from [4]).

There are various communication modules available in XBee footprint (LoRa, Bluetooth, WiFi among others), which can be used with Waspmote. Libelium offers an expansion board, which enables connecting two communication modules at the same time into the platform.

Waspmote is user-friendly platform and Libelium offers a variety of modules, but its biggest disadvantage is the price. It costs over 150€ (Waspmote board + XBee modules) and it has much less computational power than any model of Raspberry Pi.

3.2.2 XBee

XBee are RF modules produced by Digi International Inc. developed in order to meet IEEE 802.15.4 standards with the minimum power used, assuring reliable delivery of data. XBee-PRO modules are power amplified versions of XBee modules for extended-range applications. Table 3.1 presents the key features of XBee and XBee-PRO modules.

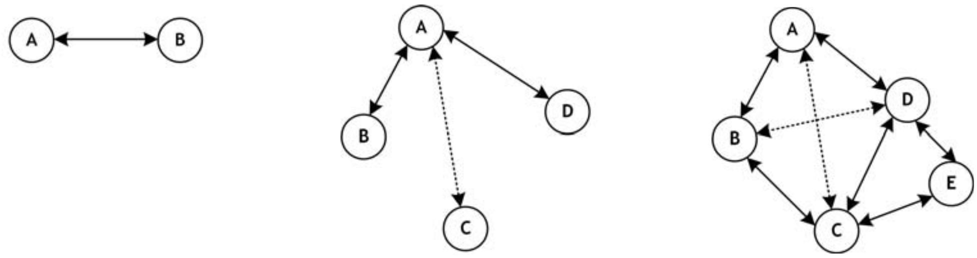
Table 3.1: XBee key features (according to [7]).

	XBee 2.4	XBee-PRO 2.4
Frequency band	ISM 2.4 GHz	ISM 2.4 GHz
Range	up to 90 m	Up to 1600 m
Transmit power	1 mW (0 dBm)	63 mW (18 dBm)
Receiver sensitivity	-92 dBm	-100 dBm
Max RF data rate	250 kbps	250 kbps
Supply voltage	2.8 - 3.4 V	2.8 - 3.4 V

Receiver sensitivity is the lowest power level, which receiver can detect as RF signal and demodulate data. It is independent from the transmitter. As the signal is propagated, power density decreases with the distance. Improvement of receivers sensitivity extends the communication range.

Digi International Inc. produces XBee modules operating in following frequency bands:

- 868 - 868.6 MHz (Europe): 1 channel with 20 kbps data rate
- 902 - 928 MHz (Americas): 10 channels with 40 kbps data rate
- 2.4 - 2.4835 GHz (worldwide): 16 channels with 250 kbps data rate



(a) Point-To-Point. (b) Point-To-Multipoint. (c) Mesh.

Figure 3.3: XBee topologies.

XBee modules support various topologies: Point-To-Point, Point-To-Multi-Point (Star) and Mesh, see Figure 3.3. In Point-To-Multipoint topology (b), node A can communicate with nodes B and D, it can not communicate with node C, which is out of range. Thus nodes have assigned roles of coordinator, router or end device in order to extend the communication range. Coordinator can hardly go to sleep mode, because it stores packets for or from end devices. There is only one coordinator in the network and it establishes the network originally. Routers act as relays between nodes and as coordinators they can never go to sleep mode. End devices usually are low-powered sensors, which can communicate only with their parent devices, either coordinator or a router. End devices do not rely data from other nodes. In Mesh topology (c) all nodes have equal roles, there is no coordinator, each one is a router. Node A can communicate with node C through node B or D. All nodes can change their modes to sleep in order to save power. In Mesh topology routes to unreachable nodes are discovered when needed, as opposed to Point-To-Multipoint, where role assignment may limit alternative routes.

XBee-PRO modules offer various firmware to their modules, which allow using different topologies such as: XBee-PRO 802.15.4 using Star topology and XBee-PRO DigiMesh using Mesh topology. Table 3.2 presents available transmission power (TX power) for both XBee and XBee-PRO.

Table 3.2: Available transmitter's power levels for XBee and XBee-PRO (according to [7]).

	XBee	XBee-PRO
Parameter	TX power level	
0	-10 dBm	10 dBm
1	-6 dBm	12 dBm
2	-4 dBm	14 dBm
3	-2 dBm	16 dBm
4	0 dBm	18 dBm

XBee modules can be configured through AT commands while they are connected via Universal Asynchronous Receiver and Transmitter (UART) as well as from X-CTU software developed by Digi International Inc. Each module has its unique MAC address, configurable node ID and configurable transmission power to several values (see Tabela 3.2). XBee modules support the 128-bit Advanced Encription Standard (AES) to encrypt data within the network.

3.2.3 LoRa

LoRa module has been developed by Semtech company. It works on two frequencies: 863-870 MHz in Europe using 8 channels with bandwidth of 0.3 MHz per channel and 902-928 in United States of America (USA) using 13 channels with a bandwidth of 2.16 MHz per channel. The key features of LoRa are presented in Table 3.3.

Table 3.3: LoRa module key features (according to [5]).

	SX1272
Frequency band	863-870 MHz
Range in LoS	up to 22 km
Range without LoS	1.5 km
Transmit power	14 dBm
Receiver sensitivity	-134 dBm
Supply voltage	3.3 V

LoRa module has adjustable power level, as it is in XBee, see Table 3.4. Receiver's sensitivity is -134 dBm.

Table 3.4: Available transmitter's power levels for LoRa module (according to [5]).

LoRa module	
Parameter	TX power level
L	0 dBm
H	7 dBm
M	14 dBm

LoRa module uses the SPI pins for communication. It was designed to be applied into network with star topology, where the nodes establish point to point connections with neighbors using parameters such as Node Address. The module does not implement any security method.

(a) (b)

Figure 3.4: (a) LoRa with 4.5 dBi antenna, (b) LoRa with 0 dBi antenna (images from [5]).

LoRa module is available with two antennas, see Figure 3.4. A special 4.5 dBi antenna enables maximum range. 0 dBi antenna have been developed smaller in order to apply it for Smart Parking. Libelium company performed communication range tests on LoRa module (in Zaragoza, Spain and in Paris, France), which are described in [5]. LoRa module range in LoS is claimed to be 22 km and around 1500 m without LoS.

3.3 Drone control

Each drone in the swarm has an evolved neural network-based controller developed by authors of [26]. They proposed a novel hybrid approach combining evolutionary robotics techniques and preprogrammed logic in order to simplify the configuration process and control. Evolutionary robotics is relatively young methodology, which appeared in 90s. It aims to develop practical controllers, which can be applied to real-life robots. Process of manual programming and tuning is very time-consuming due to complexity and number of calculations, thus evolutionary algorithms are promising alternative approach in the development of autonomous robots. Evolutionary algorithm evaluates the performance of the controller in the context of swarm-behaviour and optimizes it by tuning its parameters.

Controller development process is shown in Figure 3.5 and described in detail in [6]. There have been developed a simulator JBotEvolver [27], which enables controllers' performance assessment. The simulator is based on real measurements taken in aquatic environment and it enables to model the behaviour of robots in these conditions. However it does not model physics nor fluid dynamics, which would complicate greatly the simulation process and require expensive computations.

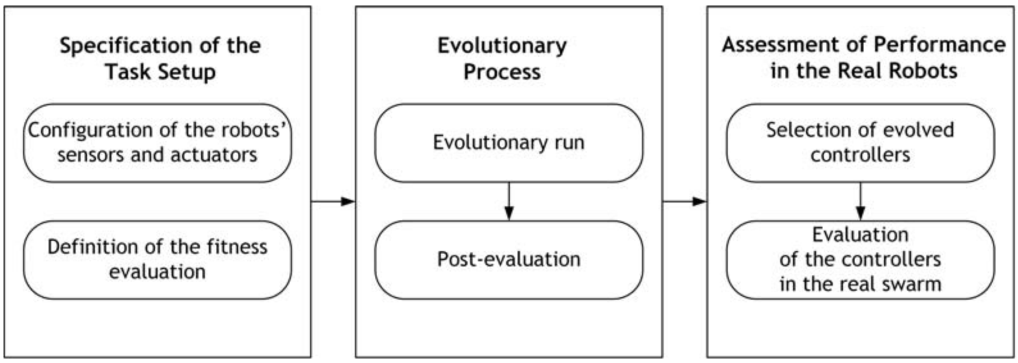


Figure 3.5: Development process of hybrid controller for each task (adapted from [6]).

The development process is run for each task separately. The configuration of sensors and actuators is an essential action, because robots depend greatly on data received from them. Next step is the definition of fitness function, which translates the objective of the task and makes it possible to see the difference between current result and desired objective. After that, the evolution process can begin. The controller is optimized by the evolutionary algorithm, and next the one, which is the best of the generation is evaluated in 100 simulations in order to precisely estimate the quality of solution. Then the next generation of controllers is created by mutations and combinations, and the process of evolution starts again. For each task, ten best performing controllers are identified based on post-evaluation results, after that three of these ten with best fitness score are selected to be tested in real robots.

3.4 Communication with and within the swarm

The communication with and within the swarm is very challenging due to high mobility of robots, which in further subsections will be called nodes. It is assumed that robots must be inexpensive and simple, so the solution is scalable. In this moment, energy consumption is not the biggest concern. The most important issue to solve is the communication with base station, which should both receive data and eventually issue some commands. In the planning process following guidelines have been considered:

- Robots are in constant highly unpredictable movement;
- Each robot has a GPS;
- Each robot transmits its position;
- The monitoring area is limited and borders are known to robots.

3.4.1 Short-range communication

The communication between robots is essential for the group success, thus the failure of one unit should not affect the global performance. There was achieved the range of 40 m in inter-robot successful communication using an IEEE 802.11g based ad-hoc wireless network (Wi-Fi). All robots broadcasted their ID, position and orientation every second exhibiting themselves to neighbors. The objective is to extend the inter-robot communication distance in order to improve the swarm performance.

The idea is to use radio frequency module such as XBee-PRO 2.4GHz, which without LoS on maximum transmitting power achieved successful communication in the distance of approximately 100 m. In distances >100m LoS has critical importance. LoS might be improved by adding mast to the robot.

XBee-PRO 2.4 GHz modules are available with mesh routing protocol called DigiMesh, which is AODV protocol based. It provides an option of multiple hops in broadcast, which might be greatly useful in swarm performance. Robots will be broadcasting their data, possibly with multiple hops.

3.4.2 Long-range technology

Some robots can carry long-range communication technology such as LoRa module, in order to communicate with the base station. Mesh networking is promising approach, which might extend the size of monitored area up to tens or hundreds of kilometers. Robots equipped with long-range technology would have special ID, which distinguish them from the other nodes. While nodes are very mobile and unpredictible, it is not recommended to implement any hierarchical protocol, thus flooding protocol might be promising solution for the communication with base station.

In flooding protocol the node, which receives flooded message m destined to another node, it rebroadcasts it only once. All other messages m received from another nodes are ignored, because a message m can be rebroadcasted only once. In this scheme it is very probable that all nodes in the network (all nodes equipped with this technology) will receive the message *m*, but there is the way of limitations such as time-to-live parameter (TTL). This parameter is defined by the transmitter and it limits the number of hops which the message *m* can jump. This parameter is included in the packet and every intermediate node decreases it by one. Thus the message *m* is rebroadcasted only when TTL value of the message is greater than zero. Flooding protocol is simple and robust scheme, especially when mobility is the main concern.

Flooding protocol is simple and robust, however authors of [28] agree that it causes a broadcast storm problem.

3.5 Heterogenous short and long-range communication and topology control for HANCAD

Since energy consumption is not a priority issue, presented topology control scheme does not involve switching modes of operation. The priority task is to maintain the communication with base station, so it could both receive data and eventually issue some commands.

In [11] the authors have suggested to deploy nodes acting as gateways to the shore. The scenario is presented on Figure 3.6. Deployment of nodes, called buoys, with both short- and long-range capabilities, and fixed localization increases chances of successful communication with the base station. The quantity of gateway nodes depends the monitored area and radio coverage.

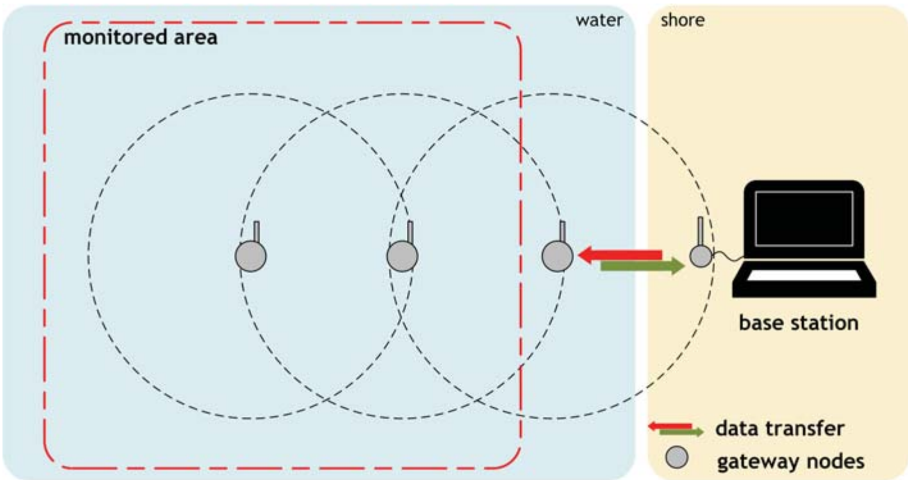


Figure 3.6: Nodes acting as gateways to the shore.

A communication within a swarm of drones is held using short-range communication technology, such as XBee-PRO devices. Let's call the network of XBee-PRO devices 'X'. It might happen that there will be no link to the base station from the swarm using X network, thus there is a need for an alternative method of delivering data. Drones can send the data to the nearest neighbor with long-range capabilities, because each one is easily recognizable by its specific ID.

It might happen that there will not be any node with long-range capabilities in the neighborhood. Each node is aware of the localization of all buoys, so in order to pass the data to the base station it must go in the direction of the closest buoy.

3.6 Summary and conclusions

In this chapter, technologies often applied to WSNs have been presented, satisfying requirements of IEEE 802.15.4 standard, which was developed to provide features such as simplicity, low-power and low-cost to WSNs. XBee is a promising technology for the swarm of aquatic drones. It is available on the market with various protocols and it is not expensive. XBee-PRO modules are available with DigiMesh protocol developed by Digi International Inc., which is based on AODV protocol. LoRa modules might be applied to the swarm of aquatic drones as a communication link from a swarm to a control station, due to their long-range capabilities.

A heterogeneous short and long-range communication solution with topology control have been proposed for a swarm of aquatic drones. A set of buoys (self-positioning vessels or anchored) with both short- and long-range communication capabilities increases greatly the communication coverage within the sensing field, because the vessels might be dispersed over the sensing field due to executed task, which might result in difficulties in local communication. The number of buoys should be adjusted to the size of monitored area, so drones could find them relatively quickly. As buoys have fixed location and every agent is aware of their location, chances of successful reporting to the base station increase, because each agent can orient himself in the direct of the closest buoy.

A scenario of autonomous mobile vessels, with both short- and long-range capabilities have been considered, which might serve as gateways to the base station, moving between drones and gathering the information from them and relaying it to the base station, as shown in Figure 1.2.

Chapter 4

Implementation of hierarchical communication solution

4.1 Introduction

Application of hierarchical solution is not a trivial task due to involvement of various technologies, which may have provided programming libraries developed by the producer or not. It may happen also that the libraries developed by the producer are not compatible with all platforms, e.g., not all libraries developed for Arduino platform are compatible with Raspberry Pi platform. Implementation process can be carefully planned and even though some unexpected problems may appear such as hardware breakdown. This chapter focuses on the implementation of communication protocol between Raspberry Pi and radio modules in C++.

4.2 Communication with XBee

XBee modules offer two modes of operation: AT and API. AT mode is called a Transparent Mode. In AT operation, any data, which are sent to XBee from the microcontroller are immediately sent to the module identified by Destination Address saved in the radio memory. In this mode the radio can be configured through the software provided by Digi International Inc. X-CTU, see Figure 4.1.

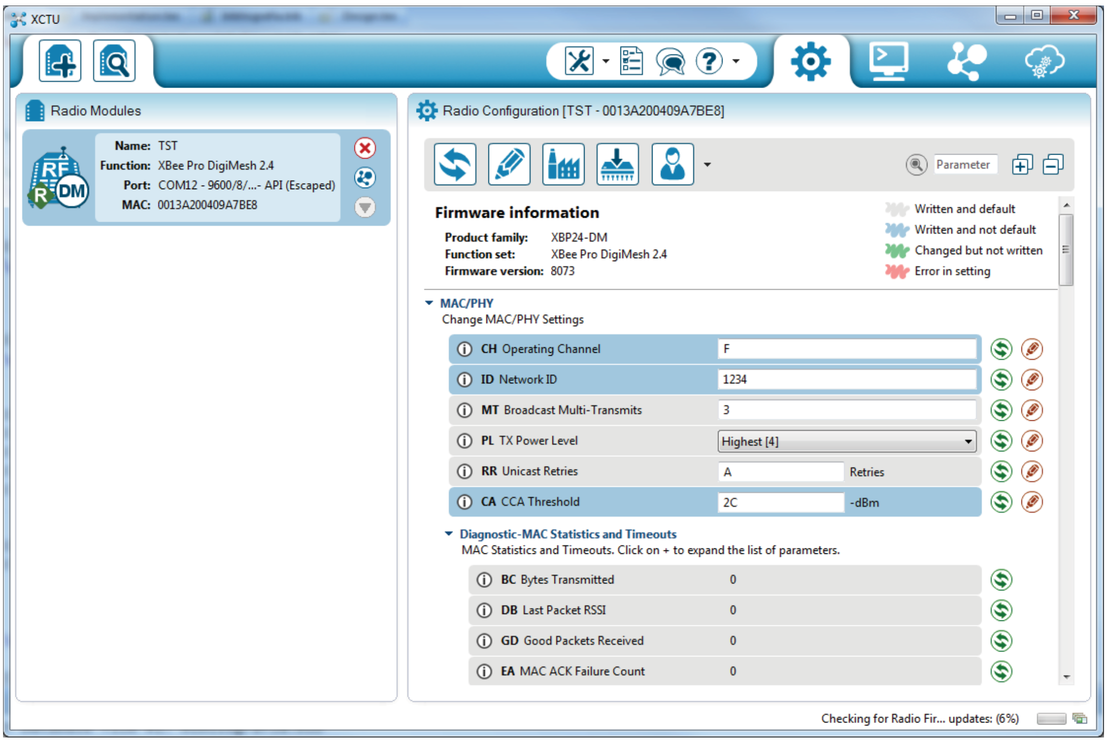


Figure 4.1: X-CTU enables both configuration and communication testing.

The alternative way of configuration is communication with the radio via UART using AT commands. The structure of AT command is presented on Figura 4.2. It always starts with "AT" prefix, it is followed by the command, which consists of two signs, e.g., NI - Node Identifier. If the user wants to write a new parameter, then the command should be followed with desired value and a carriage return in the end. If the user only wants to read the parameter, then the command should be followed with a carriage return. The list of AT commands and available parameters is can be found in [8].

In order to start the communication with the module, the user must send the 3-character command sequence such as "+++" through the serial interface within one second. Af this action is successful, the module enters the command mode and replies with OK. Since then the user can issue commands. If the user intends to change some parameters, in the end of configuration process ATWR command (ATTENTION WRITE command) should be issued, in order to save new configuration. Command mode operation can be terminated with ATCN command. The module will exit the command mode automatically if no valid command is received within the command mode timeout.

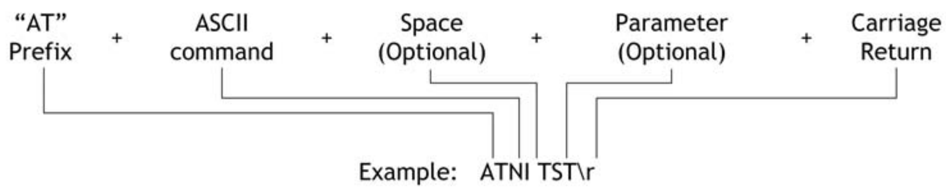


Figure 4.2: Structure of AT frame with an example.

The other method of communication with XBee module is Application Programming Interface described in following section, which involves data exchange within structured frames.

4.3 Communication between Raspberry Pi and XBee

Application Programming Interface (API) operation mode requires the communication to be managed in through the structured frames, called API frames. API mode enables a variety of operations such as:

- XBee module can be configured from a microcontroller;
- Nodes can be configured remotely;
- Data can be transmitted to multiple destinations;
- Each transmitted packet is followed by the status message (success or failure);
- Each received packet contains the source node address.

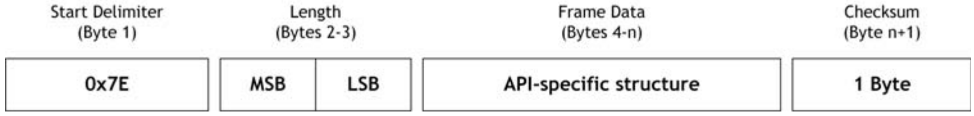


Figure 4.3: Structure of API frame.

An API frame has the structure presented in Figure 4.3.

- **Start delimiter** is a 0x7E byte and can only appear in the beginning of the frame.
- **Length** is a number of bytes in the frame between 4th byte of the frame and checksum (length nor checksum bytes are not included).
- **Frame data** field starts with API Identifier, which defines API frame type. Depending on API Identifier the content of Frame data field varies.
- **Checksum** field contains the difference between 0xFF and the sum of bytes between Length and Checksum field (the sum of bytes in frame data field).

XBee can operate in one of two API modes: API or API escaped mode. API mode runs when parameter AP = 1. In API escaped mode some specific frame data must be escaped. It was developed to prevent conflicts with special characters such as start-of-frame byte (0x7E) or flow-control characters XON (0x11) and XOFF (0x13). An Escape character 0x7D has been introduced. All previously mentioned bytes (0x7E, 0x7D, 0x11, 0x13) must be escaped by inserting 0x7D in the place of troublemaking byte and following it by the byte to be escaped XOR'd with 0x20. Escape operation can be performed from the 2nd byte of the frame (Length) until the last one (Checksum).

Example: 0x7E 0x00 0x02 0x23 **0x11** 0xCB

The byte 0x11 needs to be escaped. After the escape operation, even though 0x7D was added into the frame, the length byte does not increment.

Result: 0x7E 0x00 0x02 0x23 **0x7D 0x31** 0xCB

4.3.1 API frames

There are various types of frames supported in API mode, such as Transmit Request, Transmit Status, Receive Packet, AT Command and AT Command Response among others.

4.3.1.1 Transmit Request

Transmit Request frame tells the device to send an RF packet to another XBee module. The structure is presented on Figure 4.4.

Figure 4.4: The Transmit Request frame.

Table 4.1 presents the elements of Transmit Request frame with the example frame and description. The example frame in API mode:
7E 00 12 10 01 00 13 A2 00 40 9A 7B E8 FF FE 00 00 54 45 53 54 BF

If Escape mode is enabled (AP=2) certain bytes of the frame must be escaped. The above frame with escaping enabled looks like below:
7E 00 12 10 01 00 **7D 33** A2 00 40 9A 7B E8 FF FE 00 00 54 45 53 54 BF

4.3.1.2 Transmit Status

Transmit Status frames are sent after the Transmit Request message is completed. Their objective is to signalize the success or failure of the transmission. The structure of Transmit Status message is shown on Figure 4.5.

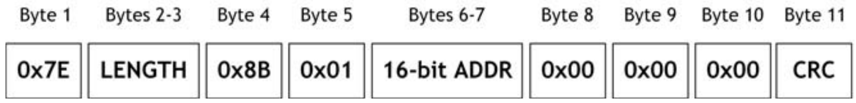


Figure 4.5: The Transmit Status frame.

Table 4.2 presents the elements of Transmit Status frame with the example frame and description. The example Transmit Status frame in API mode:
7E 00 07 8B 01 FF FE 00 00 00 76

4.3.1.3 Receive Packet

Receive Packet is a message that was sent from the other device and was received via UART. The structure of the Receive Packet is shown on Figure 4.6.

Figure 4.6: The Receive Packet.

Table 4.3 presents the elements of Receive Packet with the example frame and description. The example Receive Packet frame in API mode:
7E 00 10 90 00 13 A2 00 40 9A 7B E8 FF FE 01 54 45 53 54 3F

If Escape mode is enabled (AP=2) certain bytes of the frame will be escaped. The above frame with escaping enabled looks like below:
7E 00 10 90 00 **7D 33** A2 00 40 9A 7B E8 FF FE 01 54 45 53 54 3F

4.3.1.4 AT Command

AT Commands are used to read or write parameters of the module as well as give commands such as for example Node Discovery. The structure of Receive Packet is shown on Figure 4.7.

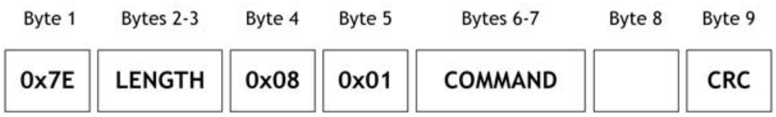


Figure 4.7: The AT Command.

Table 4.4 presents the elements of AT Commands in API mode with the example frame and description. The example API command to start Node Discovery:
7E 00 04 08 01 4E 44 64

Table 4.1: The TX Frame.

Frame field	Byte	Example	Description
Start delimiter	1	0x7E	
Length	2 3	0x00 0x12	Number of bytes between length field and checksum field
Frame type	4	0x10	Transmit Request frame type is 0x10
Frame ID	5	0x01	It identifies the UART data frame in order to correlate it with proper ACK message; if set to 0x00 there is no response
64-bit Destination Address	6 7 8 9 10 11 12 13	0x00 0x13 0xA2 0x00 0x40 0x9A 0x7B 0xE8	Each device has its own unique 64-bit address through which it can be identified; Destination Address Field should be set to 0x0000 0000 0000 FFFF in order to send a broadcast packet.
16-bit Destination Address	14 15	0xFF 0xFE	In DigiMesh protocol this field should be set to 0xFFFE
Brodcast radius	16	0x00	This field sets the maximum number of hops in broadcast transmission; if set to 0x00 then the broadcast radius is taken from maximum number of hops parameter
Transmit options	17	0x00	bit0: Disable ACK bit1: Don´t attempt route Discovery All other bits should be set to 0
RF Data	18 19 20 21	0x54 (T) 0x45 (E) 0x53 (S) 0x54 (T)	Maximum size of RF payload is defined in NP parameter. With settings BH = 0 and NH = 7, NP parameter is equal 0x49 =73 bytes. Using Encryption decreases the size of available payload.
Checksum	22	0xBF	0xFF - (sum from byte 4 to this byte)

Table 4.2: The Transmission Status Frame.

Frame field	Byte	Example	Description
Start delimiter	1	0x7E	
Length	2 3	0x00 0x07	Number of bytes between length field and checksum field
Frame type	4	0x8B	Transmit Status frame type is 0x8B
Frame ID	5	0x01	It identifies the UART data frame in order to correlate it with proper ACK message
16-bit Destination Address	6 7	0xFF 0xFE	In DigiMesh protocol this field should be set to 0xFFFE
Transmit Retry Count	8	0x00	The number of retransmission retries that occured.
Delivery Status	9	0x00	0x00 Success 0x01 MAC ACK Failure 0x21 Network ACK Failure 0x25 Route Not Found (and others available in datasheet)
Discovery Status	10	0x00	0x00 No Discovery Overhead 0x02 Route Discovery
Checksum	11	0x76	0xFF - (sum from byte 4 to this byte)

Table 4.3: The Receive Packet.

Frame field	Byte	Example	Description
Start delimiter	1	0x7E	
Length	2	0x00	Number of bytes between length field and checksum field
	3	0x10	
Frame type	4	0x90	Receive Packet frame type is 0x90
64-bit Source Address	5	0x00	Each device has its own unique 64-bit address through which it can be identified.
	6	0x13	
	7	0xA2	
	8	0x00	
	9	0x40	
	10	0x9A	
	11	0x7B	
	12	0xE8	
16-bit Source Address	13	0xFF	In DigiMesh protocol this field should be set to 0xFFFE
	14	0xFE	
Receive Options	15	0x01	0x01 Packet acknowledged 0x02 Packet is a broadcast packet.
RF Data	16	0x54 (T)	Received RF data.
	17	0x45 (E)	
	18	0x53 (S)	
	19	0x54 (T)	
Checksum	20	0x3F	0xFF - (sum from byte 4 to this byte)

Table 4.4: The AT Command.

Frame field	Byte	Example	Description
Start delimiter	1	0x7E	
Length	2	0x00	Number of bytes between length field and checksum field
	3	0x04	
Frame type	4	0x08	AT Command frame type is 0x08
Frame ID	5	0x01	
AT Command	6	0x4E (N)	ND command returns the list of neighbor nodes
	7	0x44 (D)	
Parameter Value			If empty, then it is a query to read. If any value present, it is a value to be set.
Checksum	9	0x64	0xFF - (sum from byte 4 to this byte)

4.3.1.5 AT Command Response

The AT Commands Response frame is a message sent as a response to AT Command. The structure of the AT Command Response is presented on Figure 4.8.

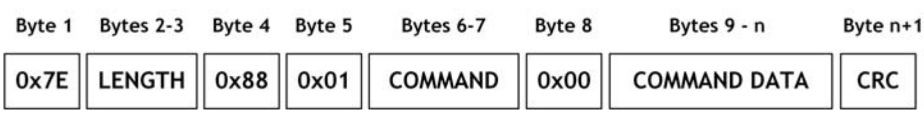


Figure 4.8: AT Command Response.

Table 4.5 presents the elements of the AT Command Response Frame in API mode with the example ND (Node Discovery) command. The API response to ND command:

7E 00 1B 88 01 4E 44 00 FF FE 00 13 A2 00 40 C1 66 1F 4C 30 31 00 FF FE 01 00 C1 05 10 1E 0D

Table 4.5: The AT Command Response.

Frame field	Byte	Example	Description
Start delimiter	1	0x7E	
Length	2 3	0x00 0x1B	Number of bytes between length field and checksum field
Frame type	4	0x88	AT Command Response frame type is 0x88
Frame ID	5	0x01	
AT Command	6 7	0x4E (N) 0x44 (D)	The frame is a response to ND command.
Command Status	8	0x00	0x00 OK 0x01 ERROR 0x02 Invalid Command 0x03 Invalid Parameter
Command Data	9 10	0xFF 0xFE	16-bit Neighbor Node Address, in DigiMesh protocol always set to 0xFFFE
	11 12 13 14 15 16 17 18	0x00 0x13 0xA2 0x00 0x40 0x91 0x57 0x2B	64-bit Neighbor Node Address
	19 20 21	0x4C (L) 0x30 (0) 0x32 (2)	Neighbor Node Identifier
	22	0x00	Reserved
	23 24	0xFF 0xFE	16-bit Parent Network Address.
	25	0x01	Device type: 0x00 Coordinator 0x01 Router 0x02 End Device
	26	0x00	Status (Reserved)
	27 28	0xC1 0x05	Profile ID: digi.com
	29 30	0x10 0x1E	Manufacturer ID: digi.com
Checksum	31	0x3F	0xFF - (sum from byte 4 to this byte)

4.3.2 Tx frame for the communication within the swarm.

A variety of data exchanged between nodes requires a defined structure of frames in order to simplify the analysis process. It is assumed that drones must exchange GPS coordinates and measurement data. In the process of development, there appeared a need for distinction between drones with long-range communication capabilities and others.

A Tx frame structure for the communication within a swarm has been prepared and it is presented on Figure 4.9. All elements of RF data field are divided by # sign. Each node has an identifier composed of one letter and 2-byte number. If it carries long-range technology, the letter is L, if not then X. 2-Byte number depends on the amount of nodes in a swarm and it is set before deployment. Next, there is a 2-byte frame number, which identifies the frame within the network, because the frame ID used in DigiMesh protocol enables frame identification only in local communication (UART). GPS coordinates and mission dependent data such as temperature measurements are located next.

Example payload: #L02#01#Latitude#Longitude#Orientation#Measurement1#.

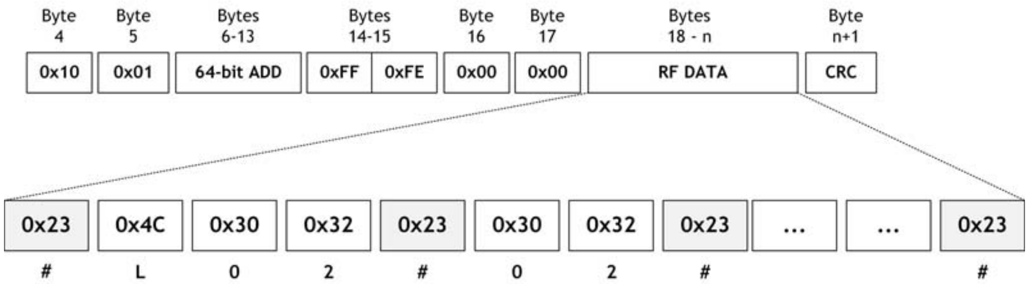


Figure 4.9: Payload of Tx frame.

4.4 Communication between Raspberry Pi and LoRa

Libelium provides the ready to use library "SX1272", which may be executed on Arduino. In order to execute it on Raspberry Pi platform, the arduPi library must be installed. It allows to use the code developed for Arduino on RPi. The module offers two types of modulation: standard FSK modulation and LoRa™ developed by Semtech. The "SX1272" library supports only LoRa™ modulation. After setting the SX1272 module ON, it is set to default LoRa™ mode.

LoRa can be configured using the Libelium's library, when the module is connected via SPI pins to Arduino or Raspberry Pi. It can be also configured, while plugged into USB LoRa Gateway and connected to the PC using terminal Putty (Windows) or Cutecom (Linux). The process of configuration through the terminal is described in [29].

LoRa has three configurable parameters: bandwidth, coding rate and spreading factor. Libelium prepared ten (1-10) modes combining different configuration of these parameters, so the module can be configured into desired mode by using one API function, see [5]. However each parameter can be configured seperately as well. The network settings such as mode or channel must be configured every time, when the module is switched on.

Some functions provided by Libelium are described in Table 4.6, the full list of available functions with descriptions can be found in [5].

4.4.1 Packet structure

The LoRa network may contain up to 254 nodes, which have unique 8-bits address. The address "0" is reserved for broadcast communication. The address "1" is reserved for the network central node. The address of destination node is given as parameter, while executing sending function.

The RF packet consists of Preamble, Header, payload and CRC, see Figure 4.10. Preamble is used to synchronize receiver with the incoming data. Header section contains Header CRC field, which indicates whether the length of payload does not exceed the maximum limit. While the preamble is detected and Header CRC does not show any error, reception process starts.

The Packet consists of the following fields: destination node address, source node address,

Table 4.6: The functionalities of the SX1272 library offered by Libelium.

Function	Description
sx1272.ON()	It opens the SPI bus and turns the SX1272 ON
sx1272.OFF()	It closes the SPI bus and turns the SX1272 OFF
sx1272.setMode()	It sets bandwidth, coding rate and spreading factor parameters
sx1272.setPower()	It sets the signal power: L - low, H - High, M - Max
sx1272.setNodeAddress()	It sets the node address in the module
sx1272.getNodeAddress()	It gets the node address in the module
sx1272.setRetries()	It sets the maximum number of retries
sx1272.getRSSIpacket()	It gets the RSSI value of the last received packet
sx1272.sendPacketTimeout()	It sends the packet to the specified destination before the timeout expires
sx1272.sendPacketTimeoutACK()	It sends the packet to the specified destination before the timeout expires and wait for an ACK response
sx1272.sendPacketTimeoutACKRetries()	It sends the packet to the specified destination before the timeout expires, wait for an ACK response and retry to send the packet if ACK is lost
sx1272.receivePacketTimeout()	It receives the information before the timeout expires
sx1272.receivePacketTimeoutACK()	It receives the information before the timeout expires and responds with ACK
sx1272.showFramefromPacket()	It prints the frame received in Packet

payload length, packet number, payload data and retry number. Destination node address is 1-byte field, editable by the user, if set to 0, the broadcast message will be sent. In broadcast message retry functionality nor ACK is available. Source node address is 1-byte field filled by the application with the module's address. Payload length is 1-byte field filled by the application, indicating total packet length. Packet number is 1-byte field filled by the application, indicating the number of a packet. It starts in 0 and when reaches 255, the value restarts. If the packet is retransmitted, this value is incremented. Data is a field for user data with the maximum size defined in the library in constant. In LoRa™ the maximum data payload is 250 bytes. The retry number is a 1-byte field, counting the number of retries. Usuall it is equal to zero, but it increments, when the packet is retransmitted. The value resets to zero after reaching the maximum number of retries or when packet was transmitted successfully.

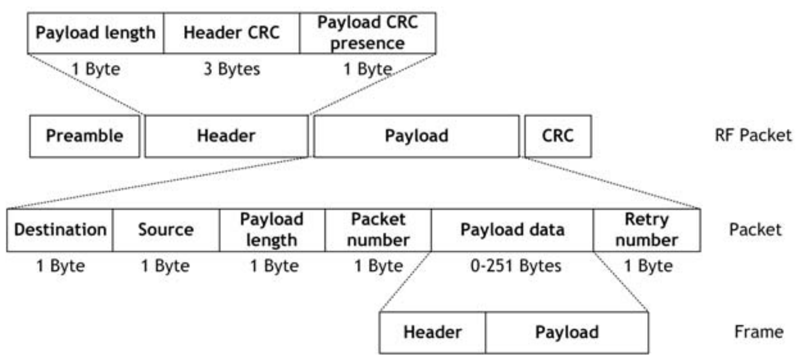


Figure 4.10: Packet structure in LoRa™.

4.5 Robot-robot communication

Robots communicate with each other within XBee network. The priority issue is the delivery of information to the base station. To ensure the delivery a communication mechanism was planned, combined of broadcasting the frame to all neighbors and transmitting frames as unicast message to nodes, which long-range capabilities in order to receive acknowledgement (broadcast frames do not have this option).

4.5.1 Broadcast

Broadcasting data such as (GPS coordinates, measurements) to the neighbors is essential for the success of the swarm. DigiMesh protocol enables multi-hop communication, even for broadcasting, while the number of hops is configurable. The prepared scheme of communication is presented on Figure 4.11. Before broadcasting, the node scans the neighborhood in order to see if any node with long-range capabilities is present. If this condition is fullfilled, broadcast will be followed by unicast transmission to one of nodes with long-range capabilities. If not, the node will perform to route discovery to the closest bouy.

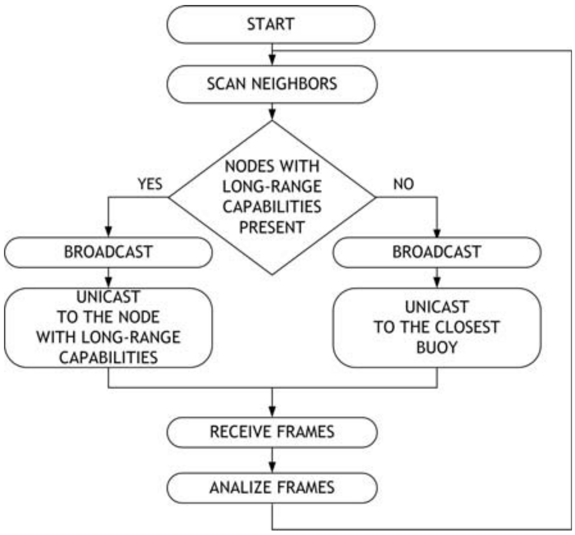


Figure 4.11: Communication scheme for robots.

4.5.2 Data analysis

When any RF data is received, the data immediately enters the DO (Data Out) buffer, from where it is sent to a connected device. The module operates in FIFO mode. When any AT command enters the DI (Data Input) buffer, AT Response will be sent to DO buffer. But it does not mean that it goes to the beginning of the line. It may happen that some RF data was received, before AT command was issued, thus these data are in the buffer before AT Response, according to FIFO mechanism. The same issue appears while waiting for Transmit Status frame after transmitting unicast message.

During tests there appeared the problem of local communication with XBee, while reading DO buffer. Sometimes it seems that the last frame is not complete, but in fact the rest of it stayed in DO buffer and while DO buffer is attempted to be read again, the continuation of the frame appears. There is no possibility to check the amount of occupied bytes in the DO buffer. In order to analyze data, all bytes coming from DO buffer are saved in file, so then they can be analized.

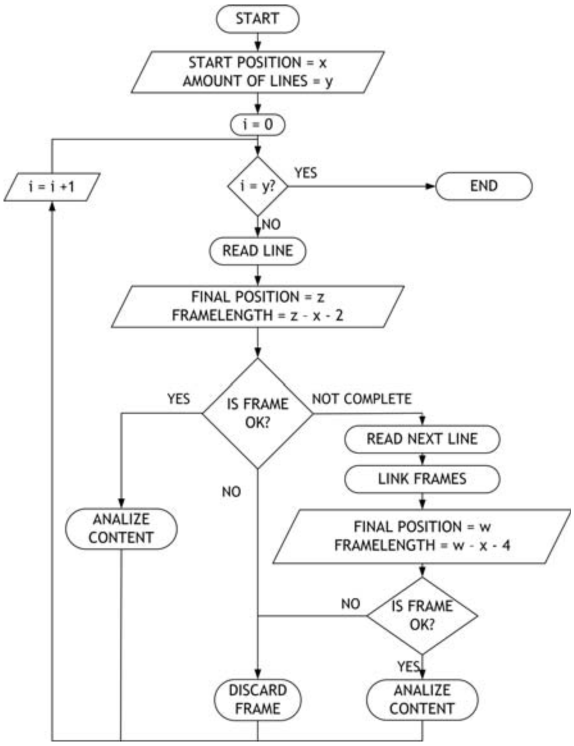


Figure 4.12: Data analysis.

4.6 Integrated hardware solution for the node with both short and long-range communication capabilities

In order to enable simple and cheap connection of two communication technologies into Raspberry Pi without involvement of USB gateways, a printed circuit has been developed. The board is simple and low-cost with basic diagnostic possibilities such as LEDs monitoring power supply , XBee mode (ON/OFF), XBee Tx line, XBee Rx line, XBee RSSI pin and XBee DI buffer. The production cost of one board is around 10 €.

Both mechanical drawing as well as pinout of XBee, SX1272 and Raspberry Pi are included in Appendix A. Pinouts of XBee-PRO and Raspberry Pi are presented in Table A.1, and Table A.2, respectively.

A circuit to connect both communication modules to Raspberry Pi has been prepared, it is shown on Figure A.2 in Appendix A. The circuit was integrated into one-layer PCB board reducing costs greatly, when compared to two-layers PCB boards. The schematic of the board is presented in Figure A.3 in Appendix A.

After printing the board, parts such as LEDs, sockets and resistors were assembled. The board is very practical, because additional cables to connect it with Raspberry Pi are not required. The board can be easily plugged and unplugged from the RPi platform. The image of the integrated circuit with assembled parts is shown on Figure 4.13. Figure 4.14 presents the board plugged into Raspberry Pi.

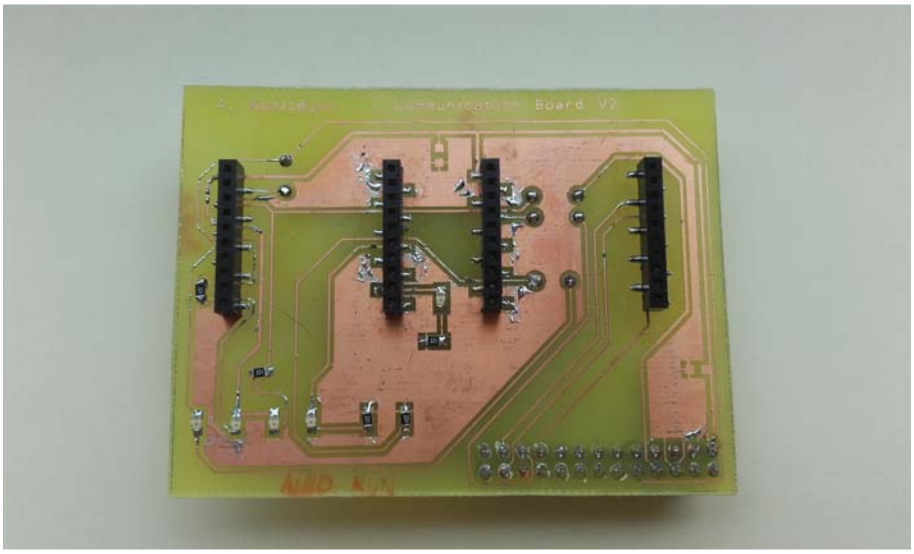


Figure 4.13: The assembled board with LEDs, sockets and a header.

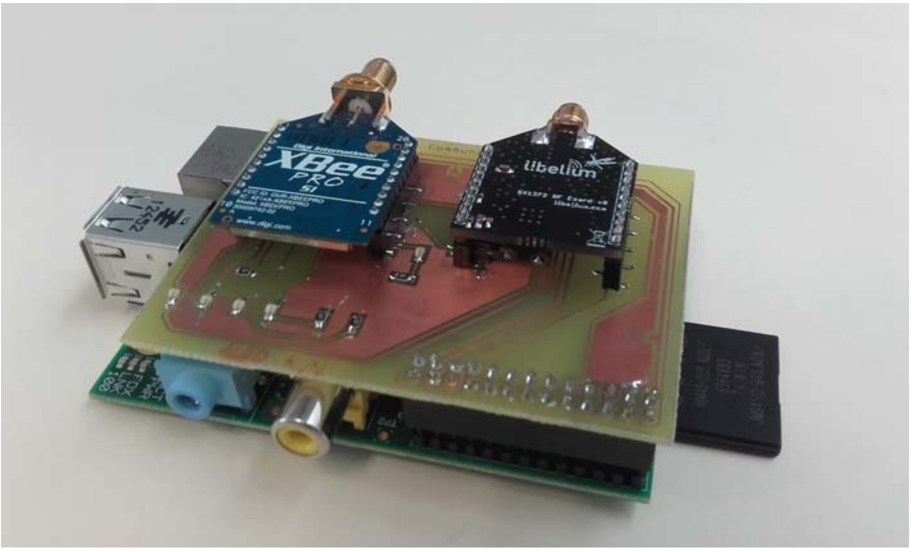


Figure 4.14: Assembled board with both radios plugged into Raspberry Pi.

The PCB was developed in order to perform tests outside the laboratory. The board is plugged into Raspberry Pi, it means that it occupies all pins of Raspberry Pi. In order to implement it to the real aquatic drone, a modification is needed to enable connection of GPS, compass, motors and sensors.

4.7 Experiment by Mondego river in Figueira da Foz, Portugal

A test between two units of Raspberry Pi with XBee and LoRa connected has been performed. The major part of the signal was distributed above the water surface. the experiment was executed by Mondego river in Figueira da Foz, the location is presented on Figure 4.15.

Figure 4.15: Location of the experiment by Mondego river.

Figure 4.16: The station with Wi-Fi connection to Raspberry Pi.

Measurement stations were placed in LoS. Their coordinates are shown in Table 4.7. Each station was composed of the laptop monitoring the transmission and one Raspberry Pi with XBee-PRO and LoRa radio plugged, see Figure 4.16. Each Raspberry Pi was powered from the powerhub. The station presented on Figure 4.16 has Wi-Fi connection with the Raspberry Pi in order to manage the operations on Raspberry Pi.

Table 4.7: Coordinates of measurement stations during the experiment by Mondego river.

Station	Latitude	Longitude
A	40°06'41.06"N	8°06'41.06"E
B	40°06'35.05"N	8°46'52.71"E

Two experiments have been run. Both nodes were broadcasting a sequence of 100 frames in the interval of 7 and 5 s using XBee. The payload of frames was: (i) #L0X#YY#Hello_7s# for 7 second interval and (ii) #L0X#YY#Hello_5s# for 5 second interval, where L0X is node identifier (station A - L01, station B - L02) and YY is frame sequence number. In the first experiment of 7 second interval, station A received 100 % of packets sent by station B, while one packet was corrupted. In the second experiment of 5 second interval, station A received 95 % of packets sent by station B due to collisions (both stations A and B were transmitting messages).

There have been also performed an experiment in order to verify range coverage according different settings of TX power on the beach in Figueira da Foz, Portugal, see Table 4.8 to see the location of stations and Figure 4.17.

Figure 4.17: The experiment was executed on the beach in Figueira da Foz, Portugal.

Table 4.8: Coordinates of measurement stations during the experiment on the beach in Figueira da Foz.

Station	Latitude	Longitude
A	40°09'42.87"N	8°52'23.19"E
B	40°08'58.69"N	8°52'06.87"E

The results of the test have been presented in [30] and can be seen in Table 4.9.

Table 4.9: The maximum coverage obtained from the experiment.

Tx power parameter	Max range [m]
Tx(0) / 10 dBm - min	65
Tx(1) / 12 dBm	120
Tx(3) / 16 dBm	1200
Tx(4) / 18 dBm - max	1400*

The maximum measured coverage distance for maximum Tx power 18 dBm (63,1 mW) with 2dBi gain antenna was 1400 m in LoS. For this Tx power, LoS has the critical importance in distances >100 m. At distances above 1400 m it was difficult to keep the LoS, so in fact the maximum range could be longer. Due to unfavourable weather conditions the test for Tx power 14 dBm was not executed.

4.8 Preliminary results on propagation model

During the experiment in Figueira da Foz, values of Received Signal Strength Indicator (RSSI) have been measured for each received packet. The settings of both transmitter and receiver are presented in Table 4.10. An appropriate place on the beach has been chosen with LoS for the experiment. The receiver was fixed on the top of wooden bench, the receiver, for each trial was moved to a proper distance, fixed on another wooden stool. Both transmitter and receiver were located on the same distance (1.2 m) above the ground, in LoS, static and their position was not changed during the transmission. The signal propagated above the straight wooden pavement surrounded by sand.

An experiment to measure RSSI values has been performed in IT-Covilhã, in University of Beira Interior in order to compare Indoor and Outdoor RSSI measurements, see Figure 4.18. In both experiments the same hardware and software settings were used. The experiment was held in a long corridor, with concrete walls on both sides and glass windows on one side. There was no object in the direct path from the signal, however few wooden benches have been present along the walls of the corridor.

Results for the received power from both experiments are presented in Tabela 4.11 and compared to Friis signal propagation model on Figure 4.19.

Table 4.10: Settings of receiver and transmitter.

Parameter	Value
Transmitter antenna gain	2 dBi
Receiver antenna gain	5 dBi
TX power level	10 dBm
RX Sensitivity	-100 dBm

Figure 4.18: Measurements of the RSSI in the corridor next to IT-Covilhã laboratory.

Table 4.11: The results of the received power, in dBm, from the indoor and outdoor experiment.

Distance [m]	1	10	15	20	25	30	35	40	45	50	55	60
Outdoor [dBm]	-65	-97	-85	-88	-104	-101	-100	-98	-96	-105	-113	-115
Indoor [dBm]	-57	-78	-66	-69	-69	-73	-73	-82	-77	-73	-79	-80

Friis propagation model describes the relation between received power and distance from the transmitter. The equation is presented in Equation 4.1. As the communication signal is an electromagnetic wave, its power is lost as it propagates. Path loss exponent represents the loss depending on the environment, it varies between 2 (perfect conditions) up to 5. Friis model presents the signal propagation in perfect conditions, thus constant parameters *C* is set to zero and path loss exponent *n* is set to 2.

$$P_r(d) = P_t + G_t + G_r + 10nlog_{10}\frac{\lambda}{4d\pi} + C \quad (4.1)$$

where

P_r : power received [dBm],
P_t : TX power [dBm],
G_t : transmitter's antenna gain [dBi],
G_r : receiver's antenna gain [dBi],
n : path loss exponent,
λ : wavelength [m],
d : distance [m]
C : system losses [dBm]

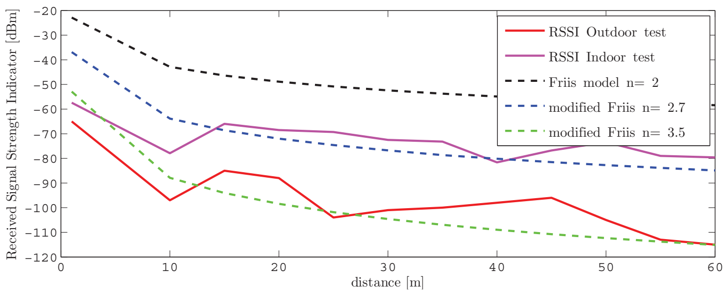


Figure 4.19: Measurements of the RSSI obtained during outdoor and indoor experiments compared to Friis model.

The loss exponent *n* = *2* used in Friis model, indicates that the communication was held through free space (air interface), in LoS, with no obstacles on the path. None of performed experiments was held in perfect conditions, in both there were appearing obstacles, such as people walking on the pavement in the outdoor experiment or walls and benches in the indoor experiment, which might have caused reflections or diffraction. Values of the RSSI obtained experimentally are different than the ones suggested by Friis model, however measurements decrease with distance as expected. They have been compared to the Friis model with modified path loss exponent *n*. In different environments path loss exponent may vary due to presence of walls or other elements.

4.9 Indoor experiment on routing performance

A test in order to determine the performance of packet delivery depending on the quality of link has been performed. Two situations have been considered, (i) node A transmitting to node B via direct link (single hop), (ii) node A transmitting to node B via one intermediate node (multi-hop). Three nodes have been used:

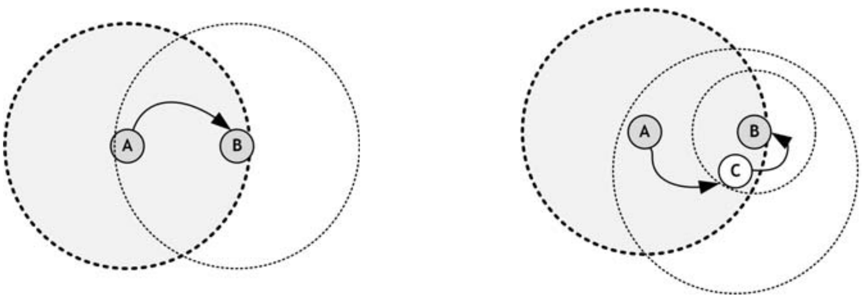
- Node A: XBee acting as transmitter, plugged through USB gateway to the Notebook, as shown in Figure 4.20;
- Node B: XBee acting as receiver, plugged through USB gateway to the PC;
- Node C: XBee plugged to Raspberry Pi via developed PCB board.

Figure 4.20: Node A transmitting.

The test was performed in the Engineering Faculty of University de Beira Interior, in IT-Covilhã. Nodes B and C were located inside the laboratory, node A was outside the laboratory, see Figure 4.22. In all tests there have been kept the distance of approximately 7 m between nodes A and B and the signal led through one wall of 0.5 m thickness. In all experiments the minimum Tx power level was used, which is PL parameter in X-CTU software (0x00, which corresponds to 10dBm) and all packets were composed of frames with maximum payload of 73 bytes, which is NP parameter in X-CTU software in XBee settings $(0x49)_{16}$.

Six time intervals have been defined for the transmission during the experiment: 100 ms, 200 ms, 500 ms, 1000 ms, 2000 ms and 5000 ms. In experiments with smaller intervals, that is 100 ms, 200 ms and 500 ms, 600 packets were transmitted. In experiments with larger intervals, greater than a second, 60 packets were transmitted. In the 1st experiment, only nodes A and B participated, see Figure 4.21a. Both had direct communication to each other. In other experiments, three nodes were combined. The antenna from node B was removed and it was verified that there was node B had no communication to node A anymore, see Figure 4.21b. Node C was located in a straight line between nodes A and B, in one meter distance from node B.

1st experiment : Single-hop connection between nodes A and B, packets are transmitted in defined time intervals;



(a) Node A transmitting to node B.

(b) Node A transmitting to node B through the intermediate node C.

Figure 4.21: Nodes ABC.

Figure 4.22: Location of node A outside the laboratory.

2nd experiment : Multi-hop connection between nodes A and B, using one intermediate node C acting as relay. Multi-hop broadcast is performed in previously mentioned intervals, from 100ms up to 5000ms;

3rd experiment : Multi-hop connection between nodes A and B, using one intermediate node C acting as relay. Multi-hop unicast is performed with acknowledgements and route discovery;

4th experiment : Multi-hop connection between nodes A and B, using one intermediate node C acting as relay. Multi-hop unicast is performed with acknowledgements, but without route discovery;

5th experiment : Multi-hop connection between nodes A and B, using one intermediate node C acting as relay. Multi-hop unicast is performed without acknowledgements, nor route discovery.

The results from the 1st and 2nd experiment are presented in Table 4.12 and Table 4.13 respectively. Results of 3rd, 4th and 5th experiment are presented in Table 4.14 and compared among each other in Figure 4.23.

Table 4.12: Results for the received packets from the UNICAST transmission via single-hop connection.

Time interval [ms]	100 ms	200 ms	500 ms	1000 ms	2000 ms	5000 ms
Received packets [%]	88 %	85 %	100 %	100 %	100 %	100 %

Table 4.13: Results for the received packets from the BROADCAST transmission via multi-hop connection.

Time interval [ms]	100 ms	200 ms	500 ms	1000 ms	2000 ms	5000 ms
Received packets [%]	87 %	100 %	99 %	100 %	100 %	100 %

Table 4.14: Results for the received packets from the UNICAST transmission via multi-hop connection (RD - Route Discovery).

Time interval [ms]	100 ms	200 ms	500 ms	1000 ms	2000 ms	5000 ms
Experiment	Received packets [%]					
with ACK and RD	2 %	2 %	5 %	10 %	20 %	12 %
with ACK, no RD	96 %	97 %	100 %	100 %	100 %	100 %
no ACK, no RD	100 %	100 %	99 %	98 %	100 %	100 %

One concludes that time interval has relevant importance on packet delivery performance. Results from the 1st experiment unicast transmission in single-hop clearly show better performance, while the radio has more time to transmit the packet. The communication held in time intervals >500 ms gave satisfying results of 100 %.

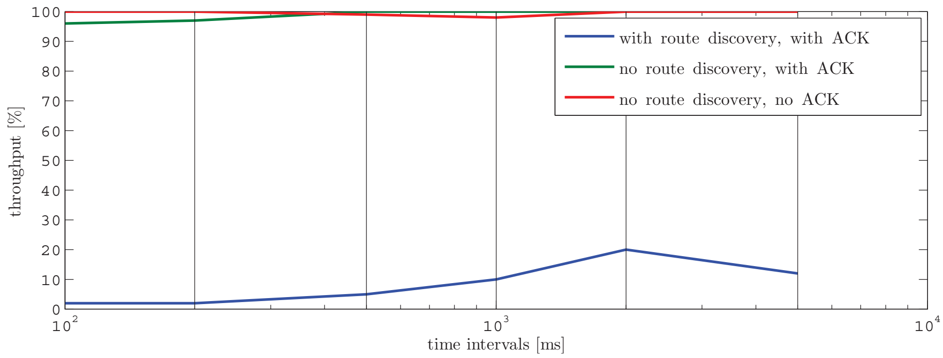


Figure 4.23: Performance of DigiMesh protocol in UNICAST transmission.

The results from unicast transmission in single-hop performance are satisfying requirements of HANCAD project. The most probable causes of packet loss in intervals <500ms are acknowledgements.

The results of 3rd, 4th and 5th experiment clearly show the impact of route discovery process and acknowledgements in UNICAST transmission. Figure 4.23 shows a clear difference in performance of AODV-based DigiMesh protocol with and without Route Discovery process running. Both route discovery process and ACK exchange involve high message exchange, causing collisions and overlapping of packets.

Broadcast operation in multi-hop performance presented much more promising results than unicast operation.

4.10 Summary and conclusions

This chapter has presented the implementation process of hierarchical solution applied to communication. Two communication technologies has been considered: XBee-PRO and LoRa. The local communication between Raspberry Pi model B and both radio modules has been described. A frame structure for the communication within the swarm, which contains all data significant for agents has been proposed.

The robot-to-robot communication scenario assumes two modes of communication, broadcast and unicast transmissions. An integrated hardware solution for the nodes with both short and long-range communication capabilities has been developed in order to perform tests outside the laboratory. Two tests have been performed, in order to determine communication range for XBee with and without LoS, they have been run in Figueira da Foz, Portugal. Results were promising, showing 100 m communication range on maximum Tx power (18 dBm) without LoS and, at least, 1400 m with LoS, possibly more.

XBee technology has been tested in terms of communication range and routing in indoor environment. The communication range met the requirements of HANCAD but routing aspects need further development. It was shown that discovery process and acknowledgements in DigiMesh protocol cause high message exchange in the network, resulting in packet delivery failure. Experiments showed a clear difference in the performance of the protocol with and without the route discovery process activated. There have not been a large difference in performance with or without acknowledgements, while route discovery is disabled.

The indoor experiment has shown promising results in broadcast in both single-hop and multi-hop communication (with one intermediate node) performance. It showed that broadcast in time intervals >200 ms is very efficient in wireless sensor network with three nodes.

An experiment in order to measure RSSI in outdoor and indoor environment has been performed. There appeared some irregularities in the scope, caused probably by obstacles present in testbed environment such as wooden benches or people walking. However, in general, the received power decreases with a distance exponentialy.

Chapter 5

Conclusions and future work

5.1 Conclusions

The key challenge of designing the communication for swarm of aquatic surface drones is to maintain connectivity between agents, so they can exchange their data in order to perform their mission well. Connectivity with base station is essential as well in order to give the possibility to human operators to monitor the performance of the swarm and eventually to issue some commands.

The mesh networking is a promising method, which gives a possibility of maintaining connectivity between the swarm and base station. The need of keeping the direct link to the base station is eliminated, probabilities of finding a multi-hop connection to the destination are high, because from assumption swarm is composed of hundreds of agents.

In this dissertation XBee-PRO and LoRa technologies have been proposed in order to be used as short and long-range communication technologies, respectively. The solution involving XBee-PRO has potential in real-world elimination, especially that one of objectives was to find a technology with communication range >30 m without LoS. The distance of 100 m meets the requirements of this task. The analysis of the performance from LoRa technology still needs more attention and testing. LoRa's frequency of 868 MHz is much lower than the frequency of XBee-PRO 2.4 GHz, thus long-range communication solution using this technology involves relatively high delays in communication.

XBee-PRO satisfies the requirement of 100 m communication range, but it is not an efficient solution for unicast transmission with route discovery process enabled. The results of experiments clearly showed the difference on routing performance in three node network during unicast and broadcast operation. The exchange of high number of messages causes failure in delivery of packets, especially if packets are generated in small intervals of time, such as <500 ms. As packet delivery during broadcast reached promising results for time intervals >200 ms, it might be the right solution for communication within the swarm.

Experiment on outdoor and indoor signal propagation had shown the difference in signal propagation, while it propagates outdoor and indoor. The curve obtained during the analysis of results, showed the decrease of power with the distance, as expected.

Raspberry Pi platform is very comfortable platform to work with, in terms of freedom of choice of various operational systems, programming languages etc. At the beginning, the code developed for communication with XBee-PRO was being developed in java environment, but due to compilation and debugging problems on Raspberry Pi platforms, decision to change the environment to C++ language was made.

Section 4.5 describes a problem of local communication between XBee platform and Raspberry Pi. As the operational system Raspbian, used in HANCAD project during development process, is Linux-based, all interruptions are handled by a kernel of the system, thus the XBee output is treated with some delay. To solve this limitation, it is worthwhile to consider to use a separated microcontroller used only to handle communication modules. The other benefit of using a separated microcontroller is the release of one communication bus of Raspberry Pi, which might be used to connect other mission dependent devices.

This dissertation contributes to hierarchical networking topic with a trial of real-world implementation of two communication technologies into a swarm of aquatic drones.

5.2 Future work

Future work involves the development of the controller, which might be implemented on a drone with XBee-PRO and LoRa technology, such as described in section 3.3. An integrated hardware solution should be developed, possibly two-layer PCB, which enables simple connection of both XBee-PRO and LoRa modules with GPS, compass, motors and other equipment to the Raspberry Pi.

Detailed signal characterization in aquatic environment of both technologies, XBee-PRO and LoRa should be performed. As it was proved that the LoS has a critical impact on communication in large distances, a development of a mast, probably telescopic one, there should be considered.

The results of the experiment had shown the impact of route discovery process on packet delivery performance in unicast multi-hop transmission, while using DigiMesh protocol based on AODV. The results of trials with route discovery process disabled were much more successful than the ones with route discovery activated. Both route discovery process and acknowledgement messages generate a high data exchange in the network causing overlapping and collisions. Broadcast performance in both single and multi-hop communication had promising results. Results for both unicast and broadcast experiments suggest that there should be investigated another type of routing protocol, which does not involve route discovery process generating so high traffic, e.g., location aware protocol.

Data analysis process could be improved, in order to minimize delays in the flow of information and to become more fault-tolerant. Topology control should be improved in order to limit the energy consumption, prolonging the life-time of each agent.

Appendix A

Integrated hardware solution for XBee-PRO and LoRa

This Appendix presents mechanical drawings of communication modules such as XBee-PRO and LoRa as well as pinouts of these modules and Raspberry Pi. It includes also the circuit and PCB board scheme developed in order to connect communication modules to Raspberry Pi.

A.1 XBee and XBee-PRO mechanical drawings

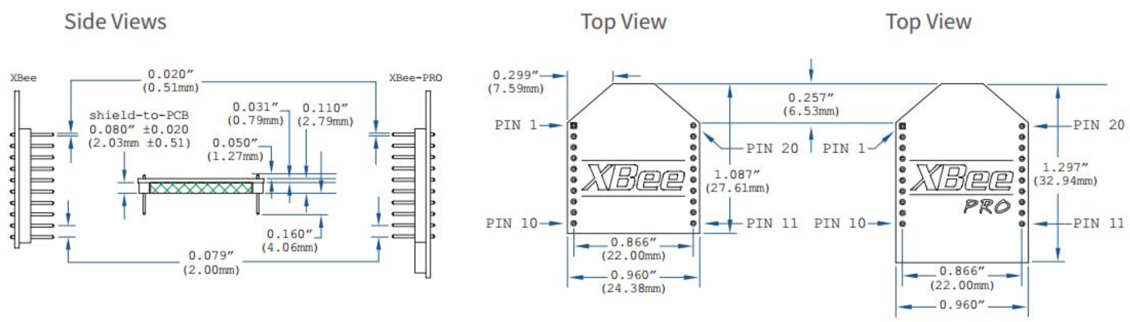


Figure A.1: XBee-PRO mechanical drawing (image from [7]).

Both XBee and XBee-PRO have the same pinout. SX1272 module has exacly the same dimensions such as XBee module, including pin spacing.

A.2 Pinout of XBee, XBee-PRO and LoRa

Table A.1 presents the pinout of both XBee and XBee-PRO radio modules.

The SX1272 module has the same pinout as XBee, but it communicates via SPI. Pins necessary to connect are:

- PIN 1 -> 3.3V
- PIN 4 -> MISO
- PIN 10 -> GND
- PIN 11 -> MOSI
- PIN 17 -> SSEL
- PIN 18 -> SCLK

Table A.1: XBee and XBee-PRO pinout (data from [8])

PIN #	Pin name	Direction	Description
PIN 1	Vcc	-	Power supply
PIN 2	DOUT	Output	UART Data Out
PIN 3	DIN/CONFIG	Input	UART Data In
PIN 4	DIO12	Either	Digital I/O 12
PIN 5	RESET	Input	Device reset
PIN 6	PWM0/RSSI/DIO10	Either	PWM output / RX signal strength indicator / Digital I/O
PIN 7	PWM/DIO11	Either	PWM output 1 / Digital I/O 11
PIN 8	Reserved	-	Do not connect
PIN 9	$\overline{\text{DTR}}$/SLEEP_RQ/DIO8	Either	Pin sleep control line or Digital I/O 8
PIN 10	GND	-	Ground
PIN 11	AD4/DIO4	Either	Analog input 4 or Digital I/O 4
PIN 12	$\overline{\text{CTS}}$/DIO7	Either	Clear-to-send flow control or Digital I/O 7
PIN 13	ON/SLEEP	Output	Device status Indicator or Digital I/O 9
PIN 14	VREF	-	Connect if you want to use analog I/O sampling. Must be between 2.6 and Vcc
PIN 15	ASC/DIO5/AD5	Either	Associated indicator, Digital I/O 5
PIN 16	$\overline{\text{RTS}}$/DIO6	Either	Request-to-send flow control or Digital I/O 6
PIN 17	AD3/DIO3	Either	Analog input 3 or Digital I/O 3
PIN 18	AD2/DIO2	Either	Analog input 2 or Digital I/O 2
PIN 19	AD1 / DIO1	Either	Analog input 1 or Digital I/O 1
PIN 20	AD0/DIO0/PB	Either	Analog input 0, Digital I/O 0 or Commissioning Pushbutton

A.3 Raspberry Pi GPIO pinout

Raspberry Pi rev. 1 was used during the development of the integrated solution. Table A.2 presents its pinout.

Table A.2: Raspberry Pi model B rev. 1 pinout

3.3 V	1	2	5 V
I2C0 SDA	3	4	Do not connect
I2C0 SCL	5	6	GROUND
GPIO 4	7	8	UART TXD
Do not connect	9	10	UART RXD
GPIO 17	11	12	GPIO 18
GPIO 21	13	14	Do not connect
GPIO 22	15	16	GPIO 23
Do not connect	17	18	GPIO 24
SP10 MOSI	19	20	Do not connect
SP10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CE0 N
Do not connect	25	26	SP10 CE1 N

Pins used in integrated hardware solution

A.4 Circuit for integrated hardware solution

There was prepared an integrated hardware solution in order to easily connect both short and long-range communication technologies such as XBee and SX1272 into Raspberry Pi. The electrical circuit is presented in Figure A.2.

Pins of XBee and SX1272 have different numeration than pins of 10x2 connectors available in the software. XBee pins 1-10 are socket's odd pins, XBee pins 11-20 are socket's even pins.

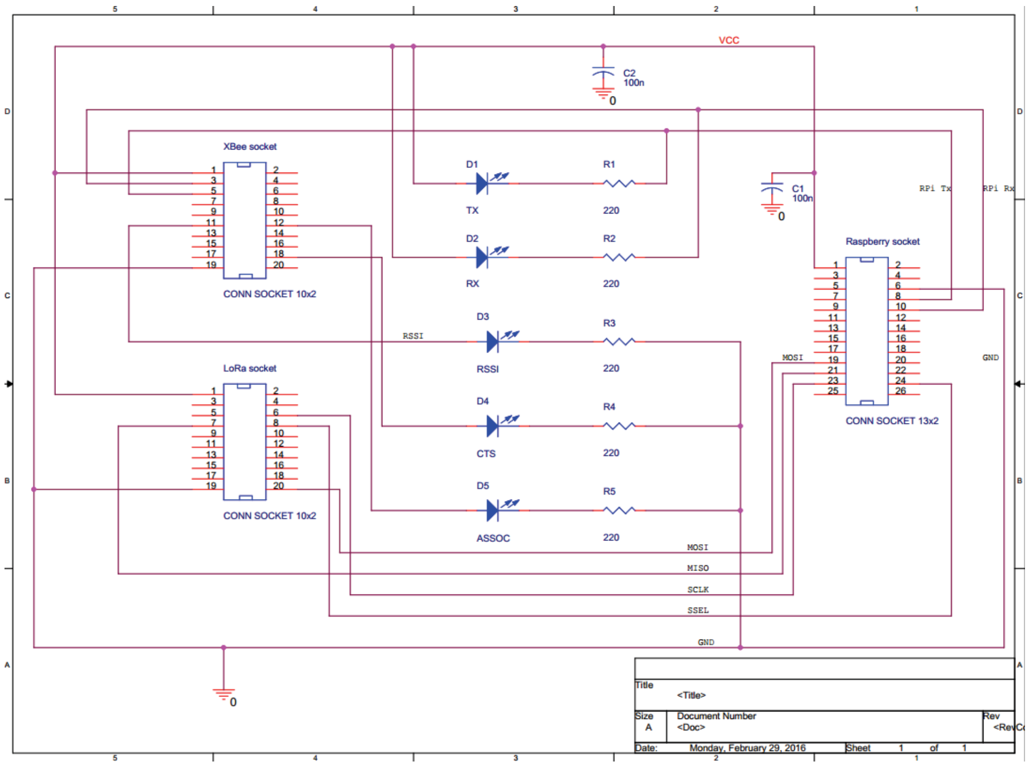


Figure A.2: The circuit developed for integrated hardware solution.

A.5 PCB board for integrated hardware solution

There have been prepared the PCB design in Orcad software implementing the circuit from previous section.

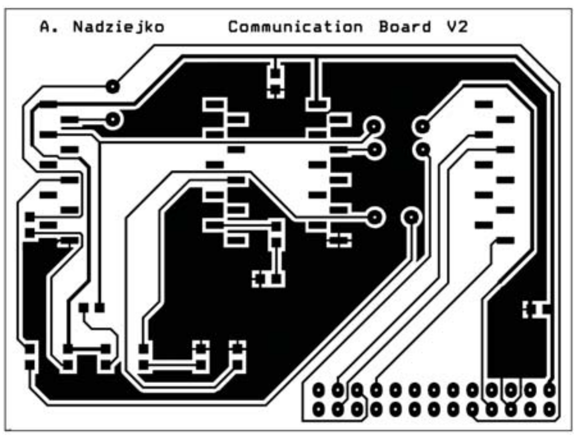


Figure A.3: The PCB.

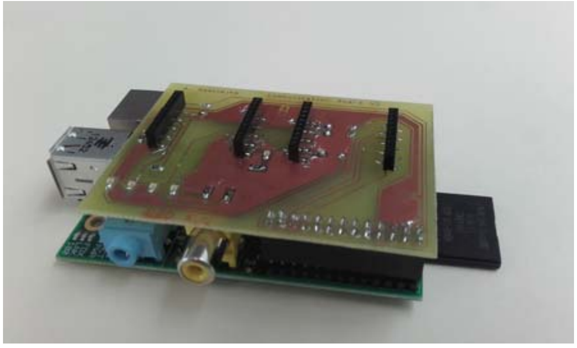


Figure A.4: The PCB board plugged into Raspberry Pi.

Appendix B

Drone Remote Console for monitoring and control of swarm of aquatic drones

B.1 Drone Remote Console

Authors of [31] have developed the software that enables both control and monitoring of the swarm. Drone Remote Console presents status information from each agent such as GPS coordinates, orientation and compass data among others as well as the image of the whole swarm on the map, see Figure B.1. Each agent's location is displayed on the map. The Console enables control of agents as well as upload of new control logic. The application can run simultaneously on various machines, enabling redundancy in case of any breakdown and multiple operators.

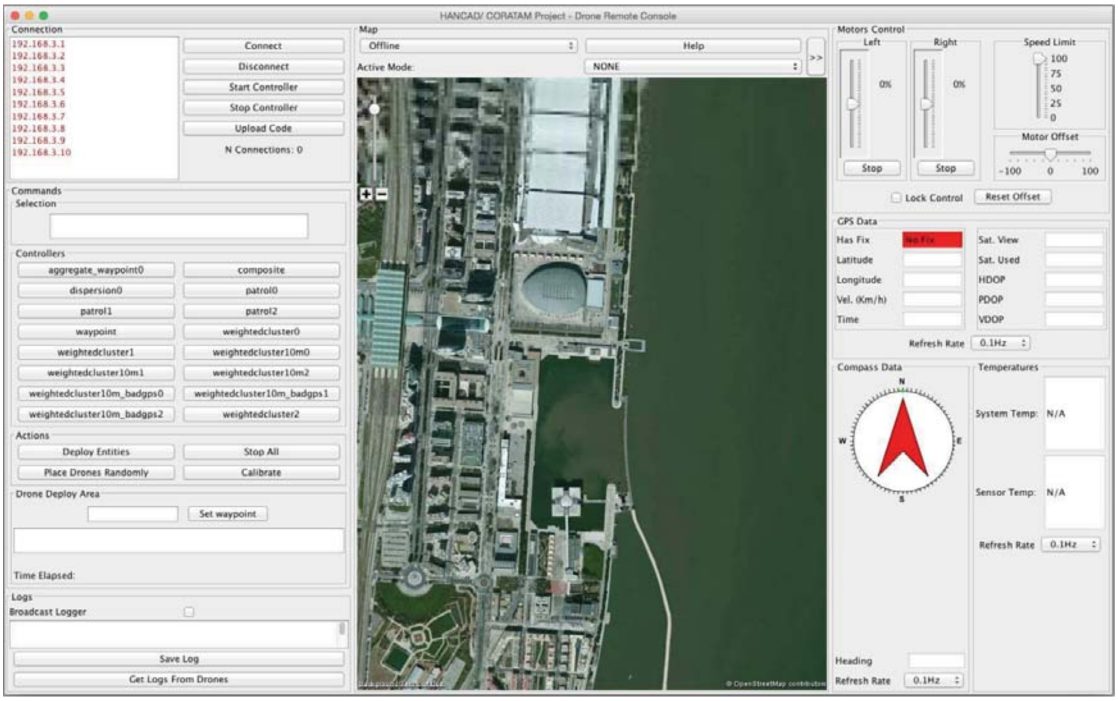


Figure B.1: Drone Remote Console.

B.2 Drone Control

Each drone has an autonomous control. The Drone Remote Station is used only to upload the control logic in the beginning of the experiment and to monitor it, see [32].

Bibliography

[1] M. A. Labrador and P. M. Wightman, *Topology Control in Wireless Sensor Networks: with a companion simulation tool for teaching and research*. Dordrecht: Springer Netherlands, 2009, ch. Topology Control, pp. 61--70. [Online]. Available: http://dx.doi.org/10.1007/978-1-4020-9585-6_6 xiii, 14, 15

[2] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, 2005. xiii, 17

[3] Biomachines Lab ISCTE-IUL. (2015) Control of aquatic drones for maritime tasks (coratam). [Online]. Available: http://biomachineslab.com/projects/control-of-aquatic-drones-for-maritime-tasks-coratam/ xiii, 19

[4] Libelium. (2014) Waspmote technical guide. [Online]. Available: http://www.libelium.com/downloads/documentation/waspmote_technical_guide.pdf xiii, 20

[5] ------. (2015) Waspmote lora 868 mhz 915 mhz sx1272 networking guide. [Online]. Available: http://www.libelium.com/development/waspmote/documentation/waspmote-lora-868mhz-915mhz-sx1272-networking-guide xiii, xv, 19, 22, 23, 34

[6] M. Duarte, V. Costa, J. Gomes, T. Rodrigues, F. Silva, S. Oliveira, and A. Christensen, ``Evolution of collective behaviors for a real swarm of aquatic surface robots," *PLoS ONE*, vol. 11, no. 3, pp. 1--25, March 2016. xiii, 23

[7] Digi Inc. (2015) Xbee digimesh 2.4 datasheet. [Online]. Available: http://www.digi.com/pdf/ds_xbeedigimesh24.pdf xiv, xv, 20, 21, 49

[8] ------. (2016) Xbee/xbee-pro digimesh 2.4 user guide. [Online]. Available: http://www.digi.com/resources/documentation/digidocs/pdfs/90000991.pdf xv, 28, 50

[9] J. E. Manley, ``Unmanned surface vehicles, 15 years of development," in *Proceedings of the MTS/IEEE OCEANS 2008 Conference*, Quebec, Canada, Sept 2008, pp. 1--4. 1

[10] T. S. Rappaport, *Wireless Communications: Principles And Practice, 2/E*. Pearson Education, 2010. [Online]. Available: https://books.google.co.uk/books?id=VmPT8B-5_tAC 1

[11] F. J. Velez, A. Nadziejko, A. L. Christensen, S. Oliveira, T. Rodrigues, V. Costa, M. Duarte, F. Silva, and J. Gomes, ``Wireless sensor and networking technologies for swarms of aquatic surface drones," in *Proceedings of IEEE 82nd Vehicular Technology Conference (VTC Fall) 2015*, September 2015, pp. 1--2. 1, 25

[12] G. Punzo, G. Dobie, D. J. Bennet, J. Jamieson, and M. Macdonald, ``Low-cost, multi-agent systems for planetary surface exploration," in *Proceedings of 63rd International Astronautical Congress 2012*, Naples, Italy, October 2012. 1

[13] J. Krause, G. D. Ruxton, and S. Krause, ``Swarm intelligence in animals and humans," *Trends in Ecology & Evolution*, vol. 25, no. 1, pp. 28--34, January 2010. [Online]. Available: http://dx.doi.org/10.1016/j.tree.2009.06.016 2

[14] C. M. Cianci, X. Raemy, J. Pugh, and A. Martinoli, ``Communication in a Swarm of Miniature Robots: The e-Puck as an Educational Tool for Swarm Robotics," in *Proceedings of Simulation of Adaptive Behavior (SAB-2006), Swarm Robotics Workshop*, ser. Lecture Notes in Computer Science (LNCS), Rome, Italy, 2007, pp. 103--115. 2

[15] M. Mihaylov, K. Tuyls, and A. Nowé, *Adaptive and Learning Agents: Second Workshop, ALA 2009, Held as Part of the AAMAS 2009 Conference in Budapest, Hungary, May 12, 2009. Revised Selected Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, ch. Decentralized Learning in Wireless Sensor Networks, pp. 60--73. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-11814-2_4 6

[16] Cisco Press. (2016) Open systems interconnection (osi) reference model. [Online]. Available: http://www.cisco.com/cpress/cc/td/cpress/fund/ith/ith01gb.htm#xtocid166844 6

[17] P. Huang, L. Xiao, S. Soltani, M. W. Mutka, and N. Xi, ``The Evolution of MAC Protocols in Wireless Sensor Networks: A Survey." *IEEE Communications Surveys and Tutorials*, vol. 15, no. 1, pp. 101--120, 2013. [Online]. Available: http://dx.doi.org/10.1109/SURV.2012.040412.00105 8

[18] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, ``Mac essentials for wireless sensor networks," *IEEE Communications Surveys Tutorials*, vol. 12, no. 2, pp. 222--248, April 2010. 8, 9

[19] C. Buratti, M. Martalo, R. Verdone, and G. Ferrari, *Sensor Networks with IEEE 802.15.4 Systems. Distributed Processing, MAC, and Connectivity*. Springer Berlin Heidelberg, 2011, ch. Wireless Sensor Networks, pp. 3--28. 10

[20] D. Gislason, *ZigBee Wireless Networking*. Newnes, 2008. [Online]. Available: http://store.elsevier.com/Zigbee-Wireless-Networking/Drew-Gislason/isbn-9780750685979/ 11

[21] A. Boukerche, B. Turgut, N. Aydin, M. Z. Ahmad, L. Bölöni, and D. Turgut, ``Routing protocols in ad hoc networks: A survey," *Computer Networks*, vol. 55, no. 13, pp. 3032--3080, 2011. 11

[22] M. Zorzi and R. R. Rao, ``Geographic random forwarding (geraf) for ad hoc and sensor networks: multihop performance," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 337--348, October 2003. 12

[23] I. Schweizer, R. Zimmermann, M. Stein, and M. Mühlhäuser, ``a-ktc: Integrating topology control into the stack," in *Proceedings of IEEE 40th Conference on Local Computer Networks (LCN), 2015*, Clearwater Beach, FL, USA, October 2015, pp. 414--417. 16

[24] J. A. Khana, H. K. Qureshia, A. Iqbalb. (2014) Energy Management in Wireless Sensor Networks: A Survey. [Online]. Available: https://www.namal.edu.pk/wp-content/uploads/2016/02/em_survey_elsjcee_14.pdf 16

[25] A. L. Christensen, S. M. Oliveira, O. Postolache, M. J. de Oliveira, S. Sargento, P. Santana, L. Nunes, F. Velez, P. Sebastiao, V. Costa, M. Duarte, J. Gomes, T. Rodrigues, and F. Silva, ``Design of communication and control for swarms of aquatic surface drones," in *Proceedings of the International Conference on Agents and Artificial Intelligence (ICAART)*. SCITEPRESS, Lisbon, Portugal, 2015, pp. 548--555. 19

[26] M. Duarte, S. Oliveira, and A. Christensen, ``Hybrid control for large swarms of aquatic drones," in *Proceedings of International Conference on the Simulation and Synthesis of Living Systems - ALIFE*, vol. 14, July 2014, pp. 1--8. 23

[27] M. Duarte, F. Silva, T. R. Rodrigues, S. Oliveira, and A. Christensen, ``Jbotevolver: A versatile simulation platform for evolutionary robotics," in *Proceedings of International Conference on the Simulation and Synthesis of Living Systems - ALIFE*, vol. 14, July 2014, pp. 1--8. 23

[28] S. Pleisch, M. Balakrishnan, K. Birman, and R. van Renesse, ``Mistral: efficient flooding in mobile ad-hoc networks," in *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing, MobiHoc '06*. New York, NY, USA: ACM Press, 2006, pp. 1--12. 25

[29] Libelium. (2016) Lora gateway tutorial. [Online]. Available: http://www.libelium.com/development/waspmote/documentation/lora-gateway-tutorial/ 34

[30] F. Velez, A. Nadziejko, A. L. Christensen, S. M. Oliveira, T. Rodrigues, V. Costa, M. Duarte, F. Silva, and J. Gomes, ``Experimental characterization of wsns applied to swarms of aquatic surface drones," in *Proceedings of the 10th Conference on Telecommunications (CONFTELE)*, 2015. 40

[31] V. Costa, M. Duarte, T. R. Rodrigues, S. Oliveira, and A. Christensen, ``Design and development of an inexpensive aquatic swarm robotics system," in *IEEE MTS Oceans Conference.*, vol. 2016, Shanghai, China, April 2016, pp. 1--8. 53

[32] M. Duarte, J. Gomes, V. Costa, S. M. Oliveira, and A. L. Christensen, *Applications of Evolutionary Computation: 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30 -- April 1, 2016, Proceedings, Part II*. Cham: Springer International Publishing, 2016, ch. Hybrid Control for a Real Swarm Robotics System in an Intruder Detection Task, pp. 213--230. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-31153-1_15 53